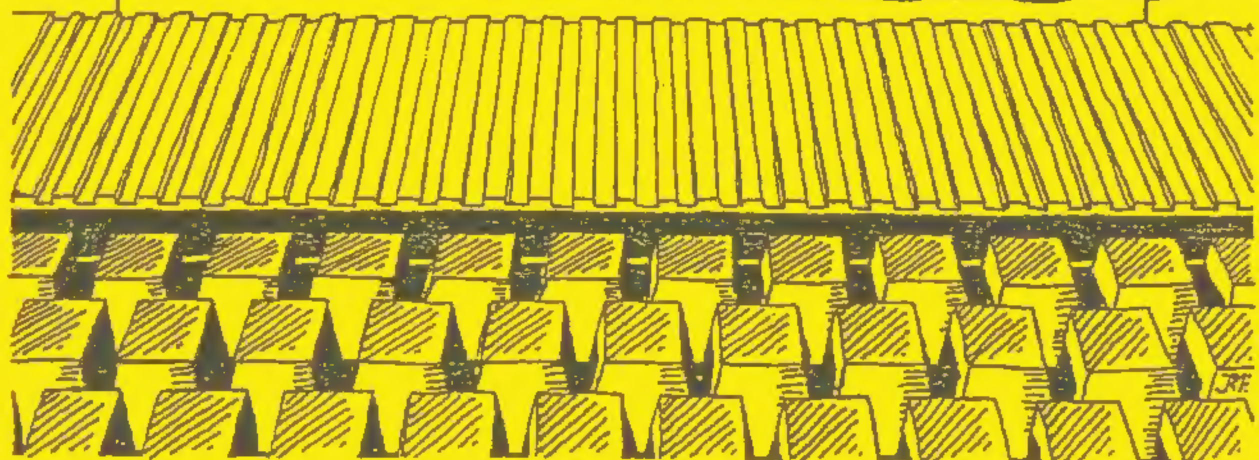
ATOM NIEUWS

JAARGANG : 8

NUMMER : 1

DISKNRS. : 1-2 1989



ATOM NIEUWS is een uitgave van de federatie Atom computerclub Ned/Belgie en verschijnt 4 keer per jaar.

VAN DE REDACTIE

Zó , beste Atommers , we hebben 1989 nu wel genoeg kans gegeven om een zinvolle start te ontwikkelen. hoogste tijd om weer eens met wat zinvolle literatuur op de markt te verschijnen en het is weer niét mis wat wij U deze keer ter verhapstukking hebben aan te bieden: 80 pagina's met voornamelijk nieuws uit ATOM-rijk.
Atom Nieuws zal, zoals is gepland, dit jaar weer viermaal verschijnen. De redactie is inmiddels weer versterkt met een niet mis te verstane aantal zwaarkalibers uit de hobbeligste hoek van Nederland, te weten Roland Leurs (bekend van zijn ongelooflijke schrijfwoede en altijd vriendelijke groeten .....) en Wil Truyen (wat niemand heeft vind je bij Wil ....)uit het bijna nederlandse maar toch nog nét Belgische koningsrijk. U vindt hun adressen op pagina 2 en het zal U daar ook opvallen dat het redactie-adres veranderd is met ingang van deze editie. Een en ander ten gevolge van taakverschuivingen binnen de redactie. Het redactieadres is voor U nu het enige verzendadres geworden, voor kopij en voor de regiosoftware. Let U a.u.b. hierop als U kopij gaat versturen en ... nu we het er toch over hebben, het is nu dus écht op. Stuurt U ons geen kopij de komende maanden , dan vrezen wij dat een volgende editie van ATOM NIEUWS op de tocht komt te staan. Dat laten wij ons natuurlijk niet aanleunen, beste Atommers, de redactie ziet dus met vol vertrouwen Uw inzendingen tegemoet. Kom op jongens, niet zeuren maar programmeuren!
Verder heeft de penningmeester ons verzocht U er op te attenderen dat de lage clubwinkelprijzen ook hun keerzijde hebben. Hij kan zich voorstellen dat U, gezien de lage prijzen, verschillende jaargangen van Atom Nieuws tegelijk wilt bestellen. Dit brengt, vooral als het hele stapels worden, portokosten met zich mee die hoger liggen dan fl. 4.-. Bij levering van "zware" bestellingen wordt U dan ook om nabetaling van porto gevraagd. Gelieve hieraan direkt te voldoen.
Wij wensen U met deze eerste editie van 1989 veel leesgenot en inspiratie voor de komende maanden.

1-4-1989

ATOM-NIEUWS REGIO-SOFTWARE '89

dnr	program	geheugen#	utility	soort	artikel	info
01	COMGess	2A00-2000	—	Utility	AN 8-1	Diskcat: HULPFILE
01	ABCON	2900-3900	P-CHARME	COMMUN.	AN.8-1	CONVERSIE ATOM<->ELECTRON
01	ABCONAS	7D00-7FFF	--	COMMUN.	AN.8-1	AECOM: CODE VOOR #7D00
01	DISINF1	2900-3268	--	UTILITY	AN.8-1	DISKINFO: PROFESSIONEEL
01	DISINF2	2900-324C	SALFAA	UTILITY	AN.8-1	DISINF1: DE SALFAA-VERSIE
01	DISKCAT	8200-920F	P-CHARME	UTILITY	AN.8-1	DISKCAT VOOR CP/M-Z80
01	FCAT	3000-7FFF	--	DATA	AN.8-1	DISKCAT: DATA-FILE
01	KLOKSRE	2900-8000	--	SPEL	AN.8-1	DIGITALE KLOK, FRAAI !
01	KNIPCUR	7000-8000	--	ZAKELYK	AN.8-1	TVMU80: HULP-FILE
01	RSCOM	2900-452F	SALFAA	COMMUN.	AN.8-1	LEES HET ARTIKEL
01	TVMMU80	2900-3987	JOSBOX	ZAKELYK	AN.8-1	TEKSTVERWERKERTJE
01	TXBOX1.	A000-AFFF	A-RAM	TOOLBOX	AN.8-1	PRINTERBOX TPX-1000
01	TXBOXSR	2900-7EBD	SALFAA	TOOLBOX	AN.8-1	TXBOX1. : DE SOURCE
01	startdc	2900-2A9D	P-CHARME	UTILITY	AN.8-1	DISKCAT: HULP-FILE
02	!BOOT	2900-2BFF	P-CHARME	SPEL	AN.8-1	DUCK: OPSTARTPROGRAMMA
02	BBC-II	4000-8000	--	OSSOFT	AN.8-1	LEES BBC<->COMBIKAART
02	BBCMOS3	3000-4000	--	OSSOFT	AN.8-1	LEES BBC<->COMBIKAART
02	CHECKER	2900-2E50	--	UTILITY	AN.8-1	UPDATE CHECKER DISK 89-12
02	COL.EMU	8200-855A		OSSOFT	AN.8-1	KLEUREN EMULATOR CLUBKRT
02	CONTROL	8000-9800	--	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	DUCK	2900-42D1	SSOFT	SPEL	AN.8-1	SPEL VOOR TOET+JOYSTICK
02	GOKKAST	2900-4E9E	--	SPEL	AN.8-1	ZEER FRAAI SPEL
02	HO+NU	9800-9E00	--	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	HO.LA	6800-8000	--	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	JVDU0.0	2900-49FF	SALFAA	OSSOFT	AN.8-1	80-KOLOMSSOFT CLUBKAART
02	MACH.T	5800-6700	-	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	MINGMAX	2900-2ABE	--	EDUCAT.	AN.8-1	BASIC 5: DEMO-FILE
02	MINIAS	2900-3753	--	OSSOFT	AN.8-1	AECOM: MINI-ASSEMBLER
02	PICTURE	8000-9800	--	SPEL	AN.8-1	GOKKST: HULP-FILE
02	PUNTEN	8000-9800	--	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	SCHERM	8000-9800	--	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	SORTEER	2900-2B4E		EDUCAT.	AN.8-1	BASIC 5: DEMO-FILE
02	SRCMAK8	8200-93FF	--	UTILITY	AN.8-1	UPDATE SOURCEMAKER
02	VDU	5000-5800	--	SPEL	AN.8-1	GOKKAST: HULP-FILE
02	WILHELM	2900-3BEA	P-CHARME	SPEL	AN.8-1	VOLKSLIED PER ATOM
02	ZOEK1	2900-2C14	--	EDUCAT.	AN.8-1	BASIC 5: DEMO-FILE
02	ZOEK2	2900-3204	--	EDUCAT.	AN.8-1	BASIC 5: DEMO-FILE

REGIO-MEDEDELINGEN.

1. Regio Den Haag.

Voorzitter : W.Kautz ,Parallelweg 2 2525NA Den Haag tel.070-639420
Secretaris : idem
Penningmeester: Th.Waayer ,LCouperusstraat 6 2274XP Voorburg tel.070-862504
Software-arch.: J.Boot ,v.Naeltwijckstr 40 2274PB Voorburg tel.070-475286
Wnd.Redact. DC: Th.Waayer (zie boven)

Lever tijdig copy in bij de redactie van DATACHECK op schijf of band ,hetzij als basic-file,hezij als text-file (ED64).

Adres regio-bijeenkomsten :Exoduskerk Berenstijnlaan 263 Den Haag
data : 28 April, 16 Juni ;zaal geopend vanaf 19.45 u.

2.Regio Oost-Brabant.

LET OP !
De regio-middag van April is verplaatst naar -27 April- a.s., in verband met Koninginnedag.

3.Regio Limburg-Belgie.

Voorz.	:E.Sanders ,Rosslag 13	6049BE	Herten	t.04750-30401
Secr.	:B.Tossaint Fatimapl 85	6214TW	Maastricht	t.043-431675
Penn.	:J.Colen Provincialew.Z 27	6438BA	Oirsbeek	t.04492-1957
	H.Bastings Termilesl 113	6229VT	Maastricht	t.043-615495
	J.Bronzwaer Plautusstr 18	6417VJ	Heerlen	t.045-719647
	R.Leurs Nattenhovprkoestr5	6129LH	Nattenhoven	t.04490-36454
	M.vLeuven Parcivallaan 14	B 2580	St.Katheliijne-Waver (B)	t.(B) 015-315082
	N.Schreurs Haardboomstr.31	6191EL	Beek (L.)	t.04490-73679
	H.de Wildt Robijnstraat 4	6412SL	Heerlen	t.045-727098

Regio-avonden :7 Apr.,5 Mei,2 Juni,7 Juli.
Demo's : Apr. uitgesteld ; Mei. RTTY ,Sept/Oct plotter/metingen nader vast te stellen : koppeling aan 2e processor.

VERBETERDE SOURCEMAKER VERSIE 8.

H.Bastings.

Waarom weer een nieuwe versie ?.

1. In versie 6 (zie A.N.7-1) is waarschijnlijk een en ander te onrechte verdwenen.

2. Er kon meer ,nog beter en consequenter.

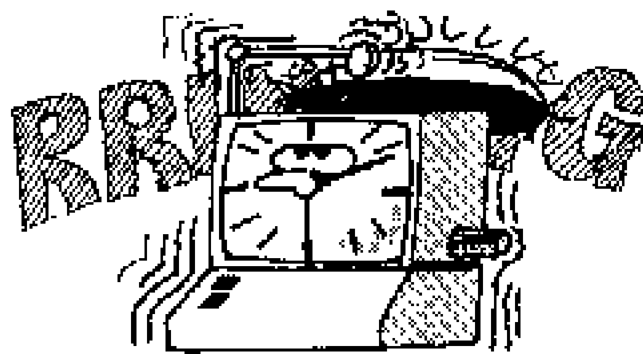
Waarvoor gebruik je Sourcemaker ?.
  a. Om een stuk machinetaal-programma te disassembleren,
  b. Om van bestaande programma's een source te maken.

Zoals een aantal van jullie waarschijnlijk wel uit eigen ervaring weten ,is het nadeel van een gewone disassembler altijd,dat zowel text alsook data-tabellen ,binnen een assemblerprogramma instaat zijn het disassembleren ernstig te bemoeilijken.
Echter uitgerekend SOURCEMAKER is in staat om binnen ATOM-programatuur in elk geval heel wat text op te sporen.
Data-tabellen zullen waarschijnlijk altijd wel moeilijk blijven, aangezien deze op geen enkele manier zijn te herkennen.

Een kleine uitbreiding is echter mogelijk,
Sourcemaker herkent namelijk het #F7D1-statement.
Dit is een veel gebruikte routine ,die de direct erna volgende karakters als text naar het scherm stuurt,totdat een waarde wordt gevonden die groter is als #00.
De DISKROM bevat echter eenzelfde routine op #E016.
De hantering van ook dit adres is in versie 8 opgenomen.

Verder is het gebruik van sourcemaker gelijk aan voorheen.
Op de regioschijf vindt U het aangepaste programma onder de titel SRCMAK8.

Veel succes ,
H.Bastings.
regio Limburg.



AECOM (ATOM-ELECTRON COMMUNICATIE)

DOOR :M.VAN WESTEN
SOFTWARE AANGEPAST
DOOR W.TRUYEN.

EEN AANTAL LEDEN VAN DE CLUB IS IN HET BEZIT VAN TWEE VERSCHILLENDE COMPUTERS VAN HET MERK ACORN.
MEESTAL ZIJN DIT EEN ATOM EN EEN ELECTRON MAAR OOK WEL EEN ATOM EN EEN BBC.
VAAK WIL MEN FILES VAN HET ENE SYSTEEM OVERSTUREN NAAR HET ANDERE BV. OM GEBRUIK TE MAKEN VAN EEN DISKDRIVE OF EEN PRINTER DIE OP HET ANDERE SYSTEEM AANGESLOTEN IS.(IN ONS GEVAL DE ATOM).
DIT IS DOOR DE VERSCHILLENDE CASSETTEFORMATEN ECHTER NIET ZONDERMEER MOGELIJK. HET PROGRAM AECOM MAAKT EEN UITBREIDING OP HET BESTAANDE COS , WAARDOOR FILES VAN HET ELECTRON/BBC FORMAAT INGELEZEN OF VERZONDEN KUNNEN WORDEN. OM HET PROGRAMMA TE KUNNEN GEBRUIKEN MOETEN ATOM EN ELECTRON DOOR EEN ZGN. KRUISKABEL MET ELKAAR VERBONDEN WORDEN. DAT WIL ZEGGEN DE CASSETTE INGANG VAN DE ATOM LIGT AAN DE UITGANG VAN DE ELECTRON EN OMGEKEERD. NA HET RUNNEN VAN HET PROGRAMMA OP DE ATOM ZIJN OP DE ATOM TWEE NIEUWE COMMANDO'S BESCHIKBAAR NL. *ELOAD EN *ESAVE.
*ELOAD MOET GEVOLGD WORDEN DOOR EEN ADRES. NA RETURN ZAL DE ATOM DE FILE DIE DOOR DE ELECTRON VERSTUURD WORDT VANAF DAT ADRES IN ZIJN GEHEUGEN OPSLAAN. *ESAVE MOET ALS VOLGT GEBRUIKT WORDEN:
*ESAVE "FILENAME" SSSS LLLL EEEE. FILENAME IS DE NAAM VAN DE FILE DIE VERSTUURD MOET WORDEN. SSSS IS HET STARTADRES IN HET GEHEUGEN VAN DE ATOM. LLLL IS HET EINDADRES PLUS EEN ! ! ! IN DE ATOM. EEEE IS HET EXECUTIEADRES EN MAG EVENTUEEL WEGGELATEN WORDEN. MET DEZE ROUTINES ZIJN HEEL LEUKE DINGEN TE DOEN. B.V. HET UITPRINTEN OP DE ATOM VAN EEN PROGRAMMA IN BBC-BASIC.
TIK IN DE ELECTRON IN DIRECT MODE IN:

```
*KEY1*SPOOL"PRINT"|M|MLIST|M*SPOOL|M
```

EN IN DE ATOM:

```
 10 REM PRINT
 20 DO
 30 *ELOAD 3000
 40 P=#3000
 50 DO;P=P+1;UNTIL ?P=13;P=P+1
 60 DO
 70 PRINT $2
 80 PRINT $P$13;P=P+LEN(P)+1
 90 UNTIL !P=#50532A3E
100 PRINT $3
110 UNTIL 0
```

EEN FILE IN DE ELECTRON KAN NU OP DE PRINTER VAN DE ATOM WORDEN AFGEDRUKT DOOR IN DE ELECTRON OP F1 TE DRUKKEN (NADAT BOVENSTAAND PROGRAMMA OP DE ATOM IS GERUND). OP DEZELFDE MANIER KUNNEN FILES VAN DE DRIVE NAAR DE ELECTRON GESTUURD WORDEN.
ZIE ATOM NIEUWS JR.5 NO.6 BLZ.105.

SOFTWARE:

```
AECOM   #2900 #3900 #C2B2
AECOMAS #7000 #7FFF #7000
MINIAS  #2900 #3753 #C2B2
```

```
 10 PROGRAM ** aecom **
 20 REM CONVERSIE-PROGRAMMA TUSSEN ACORN - atom EN electron
 30 REM 21-03 1986
 40 REM VERSIE  V1.0
 50 REM MARIEN VAN WESTEN
 60 REM LANDSTEINERHOF 58
 70 REM HOOGEVEEN
 80 REM TEL. 05280-74417
 85 REM SOFTWARE IS AANGEPAST DOOR W.TRUYEN REGIO LIMBURG.
 90 REM P-CHARME EN MINIAS NOODZAKELIJK !!
100 P.#21
110 FOR I=1 TO 2
120 P=#7D00;Z=P
130[
140:OSCLIVEC=#206;:OLDVEC=#23K
150:HEAD=#7C
160:CRCL=HEAD+21;:CRCH=HEAD+22
170:TUPD=#AE0E;:TEGB=#ADE0
180:OSNWLNE=#FFED;:SCHRIJFSTRING=#F7D1
190:OSWRCH=#FFF4;:SCHRIJFHEX=#F802
200:INIT;LDA OSCLIVEC+1;LDX OSCLIVEC;CMP @NEWOSCLI/256
210BNE INIA;CPX @NEWOSCLI&#FF;BEQ EXIT
220:INIA;STA OLDVEC+1:STX OLDVEC;LDA @NEWOSCLI&#FF
230STA OSCLIVEC;LDA @NEWOSCLI/256;STA OSCLIVEC+1
240:EXIT;RTS
250:NEWOSCLI;LDX @0;CLD
260:NVC:LDY @0;JSR #F876;DEY
270:NVA;INY;INX
280:NVF;LDA TABEL-1,X;BMI NVE;CMP #100,Y;BEQ NVA;DEX
290:NVB;INX;LDA TABEL-1,X;BPL NVB;INX;BNE NVC
300:NVE;AND @#7F;STA #CA;LDA TABEL,X
310STA #C9;CLC;LDX @0:JMP (#C9)
320:NVD;JMP (OLDVEC)
330:TABEL
340.AS"ESAVE";.DB ESAVE|#8000
350.AS"ELOAD";.DB ELOAD|#8000
360.DB NVD|#8000
370:ESAVE;JSR #F818;LDX @HEAD:JSR #FA65
380LDA @0;STA #02,X;STA #03,X
390LDX @HEAD+17;JSR #FA65
400LDX @HEAD+4;JSR #F893;PHP
410LDA @0;STA #02,X;STA #03,X
420LDA HEAD;LDX HEAD+1;PLP;BNE ESB
430STA HEAD+4;STX HEAD+5
440:ESB;STA HEAD+19;STX HEAD+20
450JSR #FA76;PHP;SEI;JSR #F85C;LDA @6;LDX @#7;STX #B002
460JSR SCHRIJFSTRING;.DB #0D0A;.AS "SAVING ";.BY #EA
470LDX @7;JSR #FB7A;LDY @8;LDA @0
480:ESC;STA HEAD+8,Y;DEY;BPL ESC
490LDA @1;STA HEAD+11
500LDA HEAD+17;BNE ESD:DEC HEAD+18
510:ESD;DEC HEAD+17;SEC
520LDA HEAD+17;SBC HEAD+19;STA HEAD+17
530LDA HEAD+18;SBC HEAD+20;STA HEAD+18
540BCS ESH;JMP #FA7D
```

```
550:ESH;LDA HEAD+18;BNE ESE
560:ESA;LDA @#80;STA HEAD+12;LDA HEAD+17
570STA HEAD+10;LDA @0;STA HEAD+11
580:ESE;JSR SAVEBLOCK
590LDA HEAD+12;CMP @#80;BNE ESF;PLP;JMP OSNWLNE
600:ESF;INC HEAD+8;BNE ESG;INC HEAD+9
610:ESG;INC HEAD+20;DEC HEAD+18;BEQ ESA;BNE ESE
620:SAVEBLOCK;LDX @36;JSR #FB83;LDA @#2A;JSR BUTAP
630LDY @0;STY CRCL;STY CRCH
640:NM;LDA (#C9),Y;CMP @13;BEQ NAMEND
650JSR BUTAP;INY;BNE NM
660:NAMEND;LDA @0;JSR BUTAP;LDY @8
670:ZH;LDA HEAD,Y;JSR BUTAP;INY;CPY @17;BNE ZH
680LDA CRCH;JSR TUPB;LDA CRCL;JSR TUPB
690LDY @0;STY CRCL;STY CRCH
700:ZD;LDA (HEAD+19),Y;JSR BUTAP;INY;CPY HEAD+10;BNE ZD
710LDA CRCH;JSR TUPB;LDA CRCL;JSR TUPB;RTS
720:ESCAPA;JMP ESCAPE
730:BUTAP;JSR TUPB
740:TELLER;PHA;LDA #B001;AND @#20;BEQ ESCAPA;PLA
750EOR CRCH;STA CRCH;LDX @8
760:LOOP;LDA CRCH;ROL A;BCC BZZ;LDA CRCH
770EOR @8;STA CRCH;LDA CRCL;EOR @#10;STA CRCL
780:BZZ;ROL CRCL;ROL CRCH;DEX;BNE LOOP;RTS
790:ELOAD
800LDX @HEAD+19;JSR #FA65;JSR #FA76;PHP;SEI
810LDY @16;LDA @#20
820:LSA;STA #140,Y;DEY;BPL LSA
830LDA @#0;STA HEAD+17;STA HEAD+18
840JSR SCHRIJFSTRING;.AS"SEARCHING";.BY #EA
850JSR OSNWLNE
860:RECEIVE;LDY @#00;STY #C3;LDA @#08;STA #C2
870:RCA;LDA #B001;AND @#20;BEQ ESCAPA;JSR #FCBD
880BCS RECEIVE;DEC #C3;BNE RCA;DEC #C2;BNE RCA
890JSR BETAG;CMP @#2A;BNE RECEIVE
900LDY @0;STY CRCL;STY CRCH
910:RH;JSR BETAG;STA #140,Y;INY;CMP @0;BNE RH;LDY @8
920:HD;JSR BETAG;STA HEAD,Y;INY;CPY @17;BNE HD
930JSR TEGB;CMP CRCH;BNE HEADERROR
940JSR TEGB;CMP CRCL;BEQ CONTA
950:HEADERROR;JSR SCHRIJFSTRING
960.DB #0D0A;.AS "HEADER ?";.DB #0D0A;.DB #EA00
970:BLOCKERROR;JSR SCHRIJFSTRING
980.DB #0D0A;.AS "BLOCK ?";.DB #0D0A;.DB #EA00
990:CONTA;LDA HEAD+8;CMP HEAD+17;BNE BLOCKERROR
1000LDY @0;STY CRCL;STY CRCH
1010:RD;JSR BETAG;STA (HEAD+19),Y;INY;CPY HEAD+10;BNE RD
1020JSR TEGB;CMP CRCH;BNE DATERROR
1030JSR TEGB;CMP CRCL;BEQ CONTB
1040:DATERROR;JSR SCHRIJFSTRING
1050.DB #0D0A;.AS "DATA ?";.DB #0D0A;.DB #EA00
1060:ESCAPE;JSR SCHRIJFSTRING
1070.DB #0D0A;.AS "ESCAPE";.DB #0D0A;.DB #EA00
1080:CONTB;INC HEAD+17;BNE CONTC;INC HEAD+18
1090:CONTC;INC HEAD+20;LDA HEAD+8;PHA;BNE CA
```

```
1100JSR SCHRIJFSTRING;.AS "LOADING";.DB #0D0A
1110.DB #0D0A;.BY #EA:LDY @0
1120:CB;LDA #140,Y;JSR OSWRCH;INY;CPY @17;BNE CB
1130:CA;LDA @8;JSR OSWRCH;JSR OSWRCH
1140PLA;JSR SCHRIJFHEX;LDA HEAD+12;BPL CTA
1150LDA @#20;JSR OSWRCH;LDA HEAD+10;BNE CUC;INC HEAD+8
1160:CUC;LDA HEAD+8;JSR SCHRIJFHEX
1170LDA HEAD+10;JSR SCHRIJFHEX
1180JSR OSNWLNE;LDA @7;JSR OSWRCH;PLP;RTS
1190:CTA;JMP RECEIVE
1200:BETAG;JSR TEGB;PHA;JSR TELLER;PLA:RTS
1210];NEXT;P.$6
1220LINK Z
1230END
```

S P S Sanders Print Service

Cassette interface print	5,00
MDCR interface print	5,00
Batterie backup printje	3,00
8K hoge geheugenprint	5.00
#E000 naar #1000 voor oude schakelkaart	3.00
omschakelprintje voor de 80k kaart	3.00
Bootstrapprint de Moor	6.00
Voedingsprint MDCR 12V	5.00
acoustische verbindings tester	5.50

Toegevoegd zijn:

Combikaart	30.00
Viaprintje Z80	10.00
8e printerbit	7.00

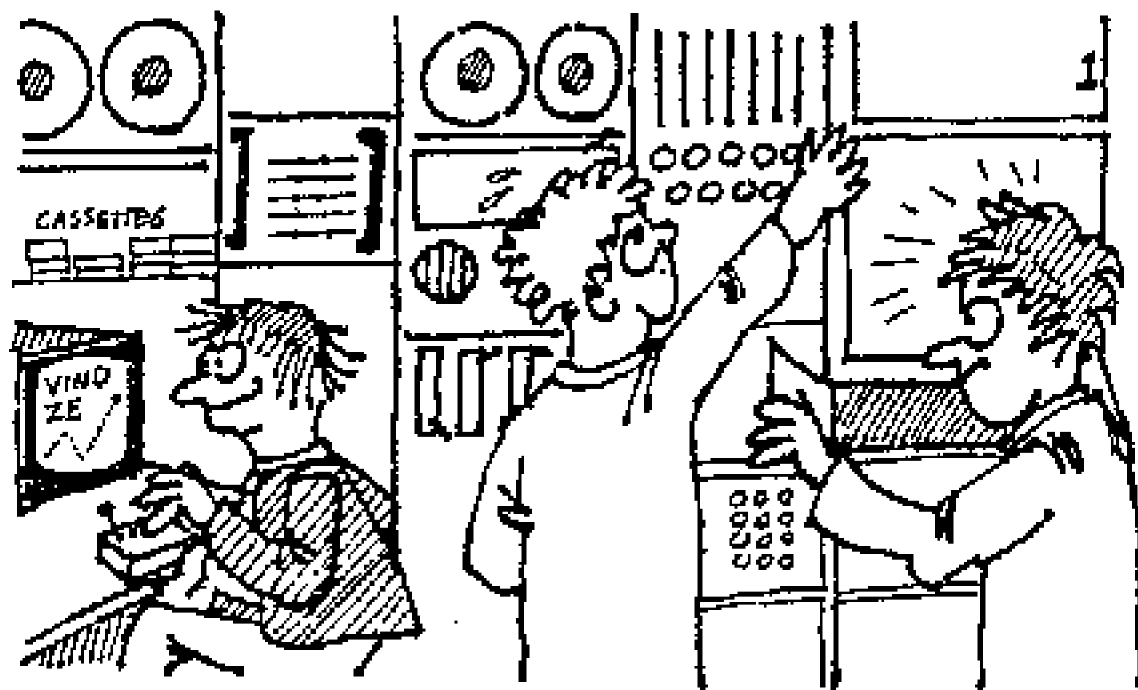
Al deze printen zijn ook gebouwd en getest te bestellen tegen kostprijs onderdelen en een symbolisch bedrag. Even bellen graag. (tel. 04750-30401)

Bestellen: op de clubavond of door overmaking van het bedrag + 1.00 + vermelding van print op gironr. 794739 tnv E. Sanders. te Serten.

Het programma DISKINFO.

Het programma DISKINFO is bedoeld als hulp voor hen die zonder veel kennis van het disk-gebeuren toch gebruik willen of moeten maken van enkele opties in bijv. de DISKROM.
Verder is het geschreven voor gebruik met het oude disk-systeem , de originele atom-controller dus.
Wil je een sector lezen van een bepaald programma dan weet je vaak niet niet waar het zich op schijf bevindt.
Door het programma in te laden en daarna diskinfo aan te roepen kun je op een makkelijke manier aan heel wat informatie komen.
Kan erg handig zijn als je programma's gesaved hebt met een of andere gekke QUALIFIER.
Bestaat deze uit een control-code , dan wordt deze vaak niet op het scherm weergegeven.
Om deze rede geeft DISKINFO de informatie in een hex-waarde en indien mogelijk ook in een ascii-karakter.
Test , gebruik of bekijk het programma . is het waardeloos voor je , dan is het vlug in de prullenbak gemikt.
Anders veel plezier ermee.

Succes , H.BASTINGS.



VAN RXBOX NAAR TXBUX

In ACORNTJESBROOD 4-1 werd een handig hulpmiddel beschreven om op wat eenvoudiger wijze met de tot wanhoop leidende ESC codes voor printers om te gaan. Ik doel op de RXBUX, ontworpen door Jan Biel en Ronald Boers.
Omdat elke printer zijn eigen eigenaardigheden heeft is deze box eigenlijk alleen goed te gebruiken voor echte EPSON-compatibele printers. Ik heb echter een TXP 1000 van General Electric, die wel veel overeenkomst met de EPSON-structuur vertoont, maar op een aantal punten toch net even anders is.
Het resultaat was, dat ik een aantal statements uit de RXBOX wel kon gebruiken, enkele niet en enkele alleen met toevoeging van enige handmatige kunstgrepen. Daar ik na verloop van tijd toch genoeg kreeg van dat gepruttel, heb ik een poging gewaagd om een voor de TXP 1000 aangepaste versie van de RXBOX te maken, wat heeft geresulteerd in, hoe raad je het, een TXBOX.
De belangrijkste veranderingen heb ik hieronder opgesomd:

De voor de TXP-1000 niet relevante statements als Italic, Emphasized, Dstrike en Unidir zijn verdwenen.

-Kingsize (80 char/line) en Language (specifieke char.set's) zijn toegevoegd.

-Enkele statements zijn van naam veranderd (Slowspeed-->Draft, RX-->TX), of worden nu voorafgegaan door TX (TXPLIST) om verwarring met in andere boxen opgenomen statements te voorkomen.

-De verschillende Condensed mogelijkheden kunnen eenduidig worden ingesteld.

-De linespacing keuze is aangepast aan de afwijkende eigenschappen van de TXP-1000 t.o.v. Epson.

-Hetzelfde geldt t.a.v. Lmargin en Rmargin, waar voor de TXP-1000 dotposities moeten worden ingevuld i.p.v. character posities, zoals bij Epson.

-De Condensed mode waarin wordt geprint bij TXMLIST is afhankelijk van het aantal opgegeven kolommen, zodat optimale leesbaarheid en papierbenutting wordt verkregen.

-In TXPLIST zijn slechts de voor de TXP-1000 van belang zijnde vragen opgenomen.

-In TX3DUMP kunnen alleen nog de geldige bit-image modes worden ingesteld.

Zowel uit de opzet van de TXBOX als de bijgevoegde beschrijving blijkt duidelijk dat ik zoveel mogelijk gebruik heb gemaakt van de oorspronkelijke RXBOX, oftewel zoveel mogelijk heb gecopieeerd. Ik wil Jan en Ronald dan ook hartelijk bedanken voor het werk dat zij voor de RXBOX hebben verricht.
Ik heb inmiddels al veel gemak van de box gehad en ik hoop dat andere TXP-1000 gebruikers er ook mee uit de voeten kunnen.

TXBOX V1.1 VOOR G.E. TXP 1000 PRINTER

A. PRINTERINSTELLINGEN:

TXBOX

Geeft het versienummer van de TXBOX, gevolgd door een overzicht van de aanwezige statements met hun entrypoint.

TXINIT

Initialiseert de TXP-1000 in de power-up conditie.
Code: ESC 24 :Initialization.

TXNORMAL

Selecteert het normale pica lettertype door het uitzetten van de volgende printmodes: condensed, elite, enlarged, specifieke character set, underline, subscript en superscript.
De line spacing blijft ongewijzigd.

TX <n> [{, <n>}]

Zet de printerdriver aan d.m.v. het sturen van STX ($2) naar de printerdriver (#FEFB) en stuurt vervolgens alle argumenten <n> naar de PRINTER (dus niet naar de printerdriver). Met TX kan elk gewenst byte naar de printer worden gestuurd, dus ook $2, $3, de inhoud van #FE en negatieve bytes.

ENLARGED <L>

Zet de TXP 1000 in de 'vaste' enlarged (double width) mode.
L=0 :Uitzetten enlarged mode.
Code: ESC W 0 :Cancel enlarged mode.
L=1 :Aanzetten enlarged mode.
Code: ESC W 1 :Set enlarged mode.

KINGSIZE <L>

Zet de TXP-1000 printmode op 60 char/line door gelijktijdig aanzetten van condensed en enlarged.
L=0 :Uitzetten kingsize mode.
Code: 18 ESC W 0 :Cancel kingsize mode.
L=1 :Aanzetten kingsize mode.
Code: 18 15 ESC W 1 :Set kingsize mode.

ELITE <t>

Schakelt de elite-pitch in of uit.
t=0 :Selecteert de pica-pitch (10 char/inch, 80 char/line).
Code: ESC P :Set pica-sized characters.
t=1 :Selecteert de elite-pitch (12 char/inch, 96 char/line).
Code: ESC M :Set elite sized characters.

CONDENSED <n>

Zet de TXP-1000 in een van de condensed (compressed) modes.
n=0 :Schakelt de condensed mode uit (evt. elite blijft aan).
Code: 18 :Cancel condensed mode.
n=1 :Zet condensed 1 mode aan (120 char/line).
Code: 18 15 :Set condensed mode.
n=2 :Zet condensed-2 mode aan (160 char/line).
Code: 15 ESC P :Set condensed-pica mode.
n=3 :Zet condensed-3 mode aan (196 char/line).
Code: 15 ESC M :Set condensed-elite mode.

SUPERSCRIPT <t>

Zet de TXP 1000 in de superscript print mode.
t=0 :Uitschakelen superscript print mode.
Code: ESC T :Cancel super-/subscript print mode.
t=1 :Inschakelen superscript print mode.
Code: ESC S 0 :Set superscript print mode.

SUBSCRIPT <t>

Zet de printer in de subscript print mode.
t=0 :Uitschakelen subscript print mode.
Code: ESC T :Cancel super-/subscript print mode.
t=1 :Inschakelen subscript print mode.
Code: ESC S 1 :Set subscript print mode.

UNDERLINE <t>

Zet de underlined print mode aan of uit.
t=0 :Uitschakelen underlined mode.
Code: ESC - 0 :Cancel underlined mode.
t=1 :Inschakelen underlined mode.
Code: ESC - 1 :Set underlined mode.

LMARGIN <n>

Stelt de linker kantlijn in op de n-de dot. Toegestane waarde voor <n> is 0 t/m 830. Expressies zijn toegestaan (b.v. n = 15 * 12 zet de linkerkantlijn op 15de character in pica mode).
Code: ESC I n1 n2 n3 :Set left margin on n1n2n3 dot position.

RMARGIN <n>

Stelt de rechter kantlijn in op de n-de dot. Toegestane waarde voor <n> is 30 t/m 960. Expressies zijn toegestaan.
Code: ESC Q n1 n2 n3 : Set right margin on n1n2n3 dot position.

LANGUAGE <n>

Kiest een van de specifieke internationale character sets.
Toegestane waarde van <n> is 0 t/m 7.

n = :	0	1	2	3	4	5	6	7
Land:	USA	France	Germ.	G.B.	D.K.	Germ.	Italy	USA

Code: ESC R <n> :Select international character set.

LSPACE <n>,<m>

Stelt de line spacing in op <n>/<m> inch.
Toegestaan zijn:

```
LSPACE 1,6        Stelt 1/6 inch line spacing in.
                  Code: ESC A 12 ESC 2
LSPACE 1,8        Stelt 1/8 inch line spacing in.
                  Code: ESC 0
LSPACE 1,9        Stelt 1/9 inch line spacing in.
                  Code: ESC 1
LSPACE <n>,72     Stelt <n>/72 inch line spacing in.
                  Code: ESC A <n> ESC 2
LSPACE <n>,216    Stelt <n>/216 inch line spacing in.
                  Code: ESC 3 <n>
```

PERF <n>

Schakelt de perforatiesprong aan of uit.

```
n=0    :Uitschakelen perforatiesprong.
        Code: ESC O  :Cancel Skip perforation.
n>0    :Inschakelen van de perforatiesprong, waarbij <n> het
        aantal lege regels is.
        Code: ESC N <n> :Set Skip perforation.
```

DRAFT <t>

Schakelt de printsnelheid van de TXP-1000 tussen Draft en Normal mode (resp. 50 en 25 char/sec).

```
t=0    :Schakelt de Normal Mode in (25 char/sec).
        Code: ESC 35  :Normal mode print speed.
t=1    :Schakelt de Draft Mode in (50 char/sec).
        Code: ESC 36  :Draft mode print speed.
```

B. PRINTER UTILITIES.

TXTAPECAT [<t>]

Geeft een overzicht op de printer van de programma's op een cassette-kant. Parameter <t> geeft aan of er een heading moet worden meegeprint.
t=0 :Geen heading printen.
t=1 :Wel heading printen.
Default is t=1, dus met heading.

In de heading worden een in te voeren titel (max. 29 char), cassettekant (A of B) en Baudrate (300 of 1200 Bd) afgedrukt. De gewenste Baudrate moet vooraf door de gebruiker worden ingesteld, b.v. met COS 1.

Van elke file worden de filename, startadres, eindadres, executieadres en aantal blokken vermeld.
Als het eerste blok (blok 0) wordt gemist, wordt het startadres berekend. dit wordt aangegeven met een sterretje voor het startadres.
Als het laatste blok wordt gemist, wordt alleen het startadres afgedrukt, gevolgd door een aantal streepjes.

TXTAPECAT kan worden gestopt met de CTRL-toets; de printer wordt dan gereset. Evt. kan met TXTAPECAT 0 worden verder gegaan, zonder dat de heading opnieuw wordt afgedrukt.

De uitdraai wordt afgedrukt in Condensed-1 mode, met 1/8 inch linespacing en zonder perforatiesprong.

Voorbeeld van het gebruik van TXTAPECAT:

```
>TXTAPECAT
ENTER HEADING:
SUPERGAMES
SIDE (A OR B):A
BAUDRATE (300 OR 1200): 1200 BAUD      (3 of 1 intypen)
PLAY TAPE

SUPERGAMES
SIDE A. 1200 BAUD
FILENAME       START END  EXEC BLOCKS
SNAPPER        2900  3C80 CE46 0013
STARGATE       2900  3A00 CE66 0011
ZOMBIE         3100  3954 C767 0008
KAMIKAZE      *2000  3C00 2000 0014
DOBBLE         2800  --        -----
```

TXTAPECAT is gebaseerd op het Cassette Catalog programma van Johan Siegers (zie Acorn Nieuws 3.2).

TXSDUMP [<fb>[,<n>[,<m>[,<ac>]]]]

TXSDUMP geeft een 'hard copy' van een zwart/wit clear 4 plaatje. Betekenis van de parameters:

<fb> - feature byte De bits 0,1 en 2 geven het volgende aan:
bit 0 : normaal/geinverteerd afdrukken.
0 = invers: een wit pixel op het scherm wordt een zwarte dot op het papier.
1 = normaal: een zwarte pixel op het scherm wordt een zwarte dot op het papier.

bit 1 : normaal/vergroot afdrukken.
0 = normale grootte (1:1)
1 = vergrote afdruk (1:2) N.B. Deze past alleen geroteerd op papier.

bit 2 : normaal/90 graden geroteerd.
0 = normale afdruk
1 = geroteerde afdruk

<n> = left margin.
<n> is het aantal spaties (12 dots) dat wordt ingesprongen vanaf de linkerkantlijn.

<m> - bit image mode TXP-1000.

m:	mode:	code:	max. dots/line
0	normal density	ESC K	480
1	dual density	ESC L	960
2	quadruple density	ESC Z	1920

Andere waarden voor <m> geven Error.

<ac> = high-byte van het beginadres van het af te drukken grafische plaatje.

Met TXSDUMP zonder parameters wordt het plaatje geprint als voor TXSDUMP 0,0,0,#80. Alleen de van deze default instelling afwijkende parameters en de voorafgaande behoeven te worden ingevuld.
Door ESC in te drukken totdat de lopende regel klaar is wordt het dumpen gestopt. De printer wordt terug gezet op 6 regels per inch en de printerdriver wordt uitgeschakeld.

De dotafmetingen op papier zijn in horizontale en verticale richting niet even groot. Hierdoor ontstaat enige vervorming, wat b.v. bij cirkels goed is te zien.
Horizontale dot afmeting is 1/60 inch = 0,0423 mm.
Verticale dot afmeting is 1/72 inch = 0.03527 mm.
Een 'normaal' plaatje wordt dus 256 * 0.04233 bij 192 * 0,03527, ofwel 10,836 bij 6,772 mm.
Een geroteerd plaatje wordt 192 * 0,04233 bij 256 * 0.03527, ofwel 8,127 bij 9,029 mm.
Vergeleken met de schermafmeting van 256 * 192 met een verhouding van 1,33 geeft een normaal plaatje een verhouding van 1,60 (+20,3 %) en een geroteerd plaatje een verhouding van 1,11 (-16,5 %) te zien. Een geroteerd plaatje geeft dus een iets betere verhouding.

Bij gebruik van de dual density mode wordt een Clear 4 plaatje half zo breed afgedrukt, waardoor het beeld ineengedrongen wordt. Voor quadruple density wordt het beeld tot 1/4 van de normale breedte gereduceerd.
Door het plaatje te vergroten wordt het effect van de dual density in horizontale richting opgeheven, maar de verticale richting is twee keer vergroot.

Als het plaatje vergroot moet worden afgedrukt, moet het tevens worden gerouleerd om op papier te passen, immers 2 x 256 = 512 dots, terwijl er max. 480 op een regel gaan. Geroteerd wordt het 2 x 192 = 384, wat dus wel gaat.

INFO [<ts>[,<n>[,<c>]]]

INFO geeft informatie over een zich ergens in het geheugen bevindend basic-programma. Achtereenvolgend worden vermeld:

- het startadres
- het eindadres
- de lengte van het programma (hex. en dec.)
- het aantal regels
- het laatste regelnummer
- de lengte van de listing in pagina's (bij benadering, omdat geen rekening wordt gehouden met regels langer dan de kolombreedte)
- de langste regel en de lengte hiervan

Indien in het programma aanwezig:
- het aantal controlcodes
- het adres van de laatste controlcode en deze code zelf
- het aantal negatieve bytes
- het adres van het laatste negatieve byte en het byte zelf

Alle bytes < #20 (spatie) + #7F (delete) worden als controlcode beschouwd. Negatieve bytes zijn alle bytes >= #80 (128).

Aan INFO kunnen drie parameters worden meegegeven:
<ts> = textspace (high-byte beginadres basic programma)

<n> = aantal regels dat op een pagina past

<c> = aantal kolommen waarin de listing wordt uitgeprint
m.b.v. het statement TXMLIST (max. 8).

INFO zonder parameters komt overeen met INFO ?18,66,1.
Alleen de van de default afwijkende parameters en de hieraan voorafgaande behoeven te worden ingevuld.

Als er in de opgegeven textspace geen programma aanwezig is volgt de melding 'NO PROGRAM'.

TXMLIST [<ts>[,<c>]]

TXMLIST geeft een Basic listing van een basic programma in een op te geven aantal kolommen, met een maximum van 8.
Bij een opgegeven aantal kolommen van 1, 2 en 3 wordt geprint in Condensed-1 (120 char/line), bij 4 en 5 kolommen in Condensed-2 (160 char/line) en bij 6, 7 en 8 kolommen in Condensed 3 (192 char/line).
Een programmaregel langer dan de kolombreedte wordt op de volgende regel in dezelfde kolom geprint, ingesprongen tot na het regelnummer.
Tijdens het printen wordt de uitvoer naar het scherm stopgezet.
Het printen kan worden afgebroken met de ESC-toets.
Er kunnen twee (optionele) parameters worden meegegeven, die de volgende betekenis hebben:

<ts> = textspace (high byte van het beginadres van het basic programma).
<c> = aantal kolommen waarin de listing moet worden geprint. Dit aantal moet zijn 1 t/m 8.

Bij TXMLIST zonder parameters wordt de huidige textspace (?18) genomen en in 2 kolommen geprint. Dit komt overeen met TXMLIST ?18,2. Alleen de van de default afwijkende parameters en voorafgaande behoeven te worden ingevuld.
Als er in de opgegeven textspace geen programma aanwezig is volgt de melding 'NO PROGRAM'.

TXPLIST

TXPLIST klapt een in compacte vorm in de TXBOX opgeslagen basic programma uit naar #8000 e.v. en runt vervolgens dit programma. Evt. geheugeninhoud op #8000 wordt dus vernietigd!!
Met TXPLIST kan op eenvoudige wijze een keurige listing van een basic programma worden afgedrukt. Er worden hiertoe achtereenvolgend (een deel van) de volgende vragen gesteld:

```
TEXT SPACE      (#29)?            Antwoorden met #82 o.i.d.
NR. COLUMS      (  1)?            Min. 1, max. 8
AUTOINDENT      ( NO)?             -+
EXTRA SPACE     ( NO)?              |  Antwoorden met Y (yes)
ELITE           ( NO)?              |               of N (no)
CONDENSED       ( NO)?            --+
LINES/INCH      (  6)?            Antwoorden met 6 of 8
PAPERLENGTH     (12")?            Antwoorden met 11 of 12

Y/N/Q/I/T?
```

Achter elke vraag wordt steeds de huidige instelling vermeld. Als dit naar wens is, is een druk op RETURN voldoende.
Na elk antwoord bepaalt TXPLIST welke zinvolle vragen er nog meer moeten worden gesteld; gekoppelde instellingen worden automatisch uitgevoerd. (B.v. bij het kiezen van meer dan 1 kolom wordt alleen nog gevraagd naar Paperlength, het afdrukken geschiedt in Condensed met 1/8 inch linespacing).

Toelichting AUTOINDENT:
Als op deze vraag "Y" is geantwoord, wordt binnen P-Charme PROCedures, FUNCTIONs, WHILE WEND blocks, CASE-END blocks, XIF-THEN-ELSE constructies, FOR-NEXT loops en DO-(F)UNTIL loops steeds met twee spaties ingesprongen.
Voorwaarde is dat er per programmaregel hooguit 1 van de bovengenoemde keywords staat; deze keywords mogen niet afgekort zijn.
AUTOINDENT werkt alleen bij listings in 1 kolom.

Toelichting EXTRA SPACE:
Door op deze vraag met "Y" te antwoorden wordt bij een listing in 1 kolom een extra spatie toegevoegd direct na het regelnummer. Dit verbetert de leesbaarheid van programma's, waarin geen spatie na het regelnummer is opgenomen.

Op de laatste vraag kan worden geantwoord met Y/N/Q/I/T.
De betekenis hiervan is als volgt:

- Y : YES — De listing wordt overeenkomstig de gekozen instelling afgedrukt op de printer. Voorafgaand aan de listing wordt eerst de naam van het programma (mits de eerste regel begint met REM of PROGRAM) en het resultaat van INFO afgedrukt.
- N : NO — Begint opnieuw bij de eerste vraag (b.v. als een vergissing is gemaakt).
- Q : QUIT — Verlaat TXPLIST en ga naar de bij de eerste vraag opgegeven text space.
- I : INFO — Voert het statement INFO uit met als parameters de gekozen instellingen (alleen op scherm, niet op printer).
- T : TEST — Geeft de listing voorzien van het resultaat van INFO en altijd in 1 kolom op het scherm weer. Hiermee kan het resultaat van AUTOINDENT vooraf worden bekeken. Met de SHIFT-toets kan de uitvoer tussentijds worden afgebroken.

Als de listing in 1 kolom wordt afgedrukt, kan het printen worden afgebroken door het indrukken van de SHIFT-toets, waarna weer om Y/N/Q/I/T wordt gevraagd.
Bij afdrukken in meerdere kolommen wordt de uitvoer via TXMLIST verzorgd; dit is te onderbreken met de ESC-toets.

Bij gebruik van losse A4 vellen dient de paperlength op 11 inch te worden ingesteld; bij gebruik van de rol kan 12 inch worden gebruikt.

TXPLIST is een voor de TXP-1000 aangepaste versie van het oorspronkelijk door Bram Pout geschreven programma PLIST.

De TXBOX gebruikt de volgende geheugenlocaties:

#70 -	#8F	:Tijdelijk gebruik bij uitvoering statements.
#140 -	#17F	:Wordt gebruikt tijdens TAPECAT en TXMLIST.
#9800 -	#9FFF	:Gebruikt door TXPLIST, incl. werkgebied en arrays.

```
*****************************************
*      H E T   W I L H E L M U S      *
*****************************************
```

door Roland Leurs

Een tijdje geleden wilde ik mijn Atom het Wilhelmus laten spelen. Er bestaat immers een programma hiervoor. Echter in het diskarchief was het Wilhelmus nergens te vinden.

Dus dat werd zoeken geblazen, om te beginnen op de regio avonden. En met de hulp van dhr. Colen kwam het programma toch nog in mijn computertje. Voor alle vaderlandslievende Atomisten laat ik het programma nogmaals afdrukken.

Met de vriendelijke groeten van Roland.

```
  0 REM ACORN NIEUWS 7
  1 REM WILHELMUS VAN NASSOUWE
  2 REM AUTEUR WIM SCHREVDER
  3 REM UIT HAREN
 10 X=-1;P.$12;RESTORE
 20 DO
 30 X=X+1
 40 P."::::WILHELMUS VAN NASSOUWE:::;"'
 50 UNTIL X=16
 60 READ X;READ Y
 70 IF X=1111 GOTO 500
 80 BEEP X,Y*12
 90 DATA 183,13,137,13,137,13,121,6,108,6,102,6,121,6
100 DATA 108,13,121,6,108,6,102,13,108,13,121,6,137,6
110 DATA 121,13,137,37,183,13,137,13,129,6,108,6
120 DATA 102,6,121,6,108,13,121,6,108,6,102,13,108,13
130 DATA 121,6,137,6,121,13,137,37,108,6,102,6,91,25
140 DATA 81,13,91,26,102,13,108,13,121,6,108,6,102,13
150 DATA 108,13,121,13,137,13,121,26,183,13,137,6,145,6
170 DATA 163,6,146,6,137,13,137,13,146,13,137,37
180 DATA 1111,0
190 GOTO 60
500 IN."TUNE NOG EENS SPELEN?(JA=1)"A
510 IF A=1 GOTO 10
520 END
```

*** JVDU-0.0 ***

INLEIDING

JVDU-0.0 is een minimale 40/80 koloms-soft voor de 80-koloms clubkaart met EF9345 CRTC. Dat "minimale" heeft betrekking op de schrijfroutine, welke uitsluitend de originele ATOM control- en karakter-codes kent. Het gebied #80..#FF is in principe (nog) niet gedefinieerd. Voorlopig hebben deze codes het afdrukken van semi-grafische mosaics tot gevolg. Er is gebruik gemaakt van de hardware-scrolling faciliteit van de 9345. Dit gaat zo waanzinnig snel, dat bij het listen van programma's een goede page mode faciliteit eigenlijk onmisbaar is. Als page mode aan staat, dan wordt bij commando en input mode bij elke aanroep van de leesroutine het aantal page regels op 24 ge-initialiseerd. Datzelfde gebeurt ook bij clear screen (P.#12) en home cursor (P.$30). Tijdens output (bijv. een listing o.i.d.) heeft de spatiebalk een "next page" functie, terwijl alle andere toetsen een "next line" functie hebben.

De leesroutine heeft wel enkele extra's t.o.v. de originele ATOM leesroutine. Dit betreft een auto-repeat faciliteit met een goede anti-dender voorziening. De auto-repeat treedt in werking als een toets langer dan 1/3 seconde ingedrukt gehouden wordt. Bij het gebruiken van de repeat toets vervalt de 1/3 seconde wachttijd terwijl de repeteersnelheid verhoogd wordt. Ook is er een voorziening aangebracht welke het maximale aantal karakters voor het commando-buffer van 63 verhoogt naar 125. Dit is op een verantwoorde manier gedaan, zie later. Hetzelfde geldt voor het input-buffer. Pas echter op met Input's van meer dan 63 karakters, want dan wordt de #180 grens overschreden. Het gebied vanaf #180 wordt o.a. door P-charme af en toe (tijdelijk) gebruikt. Control codes worden niet in de genoemde buffers opgeslagen, maar wel uitgevoerd.

Geladen vanaf #2900 loopt de source van de JVDU-soft tot #4949. Voor het assembleren is SALFAA 2.5 nodig. De object code is 3/4 K lang. De soft gebruikt in page zero uitsluitend locaties welke ook bij de standaard ATOM VDU-soft gebruikt werden, zie later.

TECHNISCHE ASPECTEN

Zoals gezegd worden in de leesroutine enkele zaken m.b.t. commando- en input-buffer opgevangen en gecorrigeerd. BELANGRIJK: DIT MAG ALLEEN ALS DE LEESROUTINE VANUIT DE INPUTROUTINE OP #CDOF AANGEROEPEN WORDT!
Bij de leesroutine in de JVDU-soft wordt dit dan ook terdege gecontroleerd. Dit is naar mijn mening de enige juiste en veilige manier om via de leesroutine de tekortkomingen van de inputroutine op te vangen.
In page zero worden de locaties #DE t/m #E7 door de soft gebruikt. Hoewel het in principe niet netjes is om vanuit BASIC en/of machinetaal rechtstreeks deze locaties te raadplegen c.q te modificeren, is dat helaas in het verleden op grote schaal toch gebeurd en dat gebeurt nog. Om al die software niet in een klap waardeloos te maken zijn we genoodzaakt enkele van deze locaties voor eens en voor altijd een vaste functie te geven:

#E0 = CURPOS : huidige cursorpositie op regel (0..79)
#E4 = XSAVE : bewaarplaats X-register tijdens lees- en schrijf-routines
#E5 = YSAVE : bewaarplaats Y-register tijdens lees- en schrijf-routines
#E7 = LMASK : lock masker (#00 of #60)

In dit rijtje zou men ook #E1 als cursorbyte verwachten. Immers om de cursor aan/uit te zetten, werden de waarden #80/#00 altijd rechtstreeks naar deze locatie gepoked. Bij deze simpele minimale JVDU-soft heb ik er voor gezorgd dat die manier nog wel werkt, zij het met de beperking dat dit alleen geldt voor een vast blokje als cursor. Deze werkwijze houdt in dat in principe 7 bits verspild worden, welke we bij een uitgebreidere VDU-soft nu juist zo hard nodig hebben.

VDU BESTURING VIA OFFICIEEL KANAAL

Acorn heeft dus helaas verzuimd om bepaalde zaken, zoals bv. het aan/uit zetten van de cursor, op een fatsoenlijke manier te regelen. Het overstappen op een andere VDU-soft wordt daardoor bemoeilijkt. Juist dat overstappen op een nieuwe VDU-soft is naar mijn mening een unieke gelegenheid om daar wat aan te doen. Sommige BASIC programmeurs denken dit op te lossen door gewoon enkele extra statements te introduceren, zoals bv. CON en COFF. Het punt is dat we dan alleen een oplossing vanuit de BASIC omgeving hebben. Zodra we ons hierbuiten bevinden (machinetaal of andere programmeertaal o.i.d.), dan zijn we nog steeds aangewezen op het afkeurenswaardige direct poken naar systeem locaties.

De enige echte oplossing is definiereing op het niveau van het Operating Systen. Het lijkt mij het beste om daarin de Acorn filosofie te volgen. Dat houdt in dat het OSWRCH kanaal behalve als karakter output kanaal ook als device besturings kanaal gebruikt wordt. Concreet houdt dit in dat we control codes uit het gebied #00..#1F voor besturingsdoeleinden gebruiken.

ALGEMEEN TOEPASBARE BESTURINGSTAKEN

Naar mijn mening komen in eerste instantie algemeen toepasbare besturings taken voor implementatie in aanmerking. Algemeen toepasbaar wil zeggen dat die taken voor elk soort VDU bruikbaar en nuttig zijn, ongeacht de gebruikte CRT-controller. Enkele voorbeelden ter illustratie:

Zonder parameters:

- INIT : initialisering van de CRTC
- CLC : clear screen vanaf cursor en lager
- CLN : clear line vanaf cursor tot einde regel

Met parameters:

- MODE X : zet CRTC in mode X (40/80 kar/regel)
- CURSOR X : zet cursor aan (X=1) of uit (X=0)
- RDCHAR A : lees kaakter op cursorpositie in A
- PUTCURS X,Y : plaats cursor op positie X,Y (pos X op regel Y)
- GETCURS X,Y : lees cursorpositie in X,Y
- PUTCHAR (S),X,Y : plaats karakter (S) op positie X,Y (Stack-item)
- GETCHAR A,X,Y : lees karakter op positie X,Y in A
- SCROLL-UP X,Y : scroll regels X t/m Y 1 regel omhoog
- SCROLL-DOWN X,Y : scroll regels X t/m Y 1 regel omlaag

Bij deze voorbeelden ben ik uitgegaan van parallelle parameter overdracht via de 6502 registers A,X,Y. Het kan nog universeler door enkele geheugen lokaties als parameter medium te gebruiken. Nog universeler is seriele parameter overdracht, maar dan is slechts eenrichtingsverkeer mogelijk. Bovendien wordt de soft dan nogal ingewikkeld.

Het inlezen van gegevens via een output kanaal komt misschien een beetje vreemd over, maar is in principe toch vrij eenvoudig te realiseren. Laten we ons echter eerst maar eens beperken tot ons eenvoudige cursor-voorbeeld. De cursor uitzetten zou nu in BASIC gaan met X=0;P.$cursor (cursor is een CTRL-code). In machinetaal zal dat er als volgt uitzien: LDX @0 ; LDA @cursor ; JSR oswrch (oswrch = #FFF4). Een COFF statement uitbreiding dat volgens deze manier geprogrammeerd is, is zodoende VDU onafhankelijk. Het enige dat nu nog VDU afhankelijk is, is de VDU-soft zelf waarin de gewenste handeling verricht wordt die bij de betreffende control code hoort.

SPECIFIEKE BESTURINGSTAKEN

Dit zijn besturingstaken welke vooral betrekking hebben op de specifieke mogelijkheden van een bepaalde CRTC. Software welke hiervan gebruik maakt is per definitie minder goed of geheel niet uitwisselbaar t.a.v systemen met verschillende CRTC's. Hoewel op zich niet minder belangrijk, zullen deze taken naar mijn mening toch een aparte plaats in moeten nemen t.o.v. de meer algemene taken. Een paar voorbeelden voor de EF9345 CRTC:

- CURS X : cursor X (0..3) blokje/underscore vast/knipperend
- PAGE X : selecteer beeldschermpagina X (0..3)
- SERVLINE X : service line aan/uit zetten (X=1/0)
- SERVCURS X : zet cursor op service line op positie X
- SERVPUTCH Y,X : plaats karakter Y op service line op positie X
- SERVSHOWCURPOS X : plaats huidige cursorpositie in de vorm van een tweecijferig getal op pos X van de serviceline.
- SHOWCURPOS X : enable/disable SERVSHOWCURPOS (X=1/0)
- NEGATIVE X : zet geinverteerd video aan/uit (X=1/0)
- FLASH X : zet knipperen aan/uit (X=1/0)
- UNDERLINE X : zet onderstrepen aan/uit (X=1/0)

VRIJE CONTROL CODES

Een probleem is dat het aantal nog vrije control codes beperkt is. Uitgaande van de originele ATOM control codes, zijn er nog 16 codes vrij in het gebied #00..#1F. Helaas hebben de ontwerpers van de huidige VDU-soft's het merendeel van deze codes toegewezen aan specifieke zaken gericht op de 9345 CRTC. Ze hebben zich daarbij naar mijn mening blind gestaard op de vele vaak exotische mogelijkheden van deze controller. De soft moest en zou toch vooral maar zoveel mogelijk allerlei mogelijkheden van deze chip benutten. Kostbare control codes zijn toegewezen aan "toeters en bellen" zoals dubbel breed, dubbel hoog, kleuren, knipperen, graphic's speciale karakters enz. Men is compleet voorbij gegaan aan algemene en elementaire zaken zoals het aan/uit zetten van de cursor e.d. Ik betreur die gang van zaken zeer. Men moet eerst zorgen dat men fatsoenlijk kan lopen alvorens te gaan rennen.

VDU-2.8 EN VDU 3.3

Ontwerpers van de huidige soft's zoals VDU-2.8 en VDU 3.3 hebben pioniers werk verricht en dat mogen we vooral niet onderschatten. Toch blijf ik erbij dat te veel hooi ineens op de vork genomen is. VDU 3.3 en bijbehorende statement uitbreidings soft staat bol van de mogelijkheden, maar helaas ook bol van de fouten (sorry, maar dat moet gezegd kunnen worden vind ik), VDU-2.8 werkt redelijk (wat waar is, is waar) maar is nogal

inefficient geprogrammeerd (1.25 K lang en enigszins traag). Beide soft's gaan echter in de fout bij het verhogen van het maximale aantal karakters voor de buffer input routine. Er wordt verzuimd om te controleren of de aanroep van de leesroutine daadwerkelijk vanuit de buffer input routine op #CD0F komt. Als OSRDCH (#FFE3) ergens anders vandaan aangeroepen wordt, dan gebeuren er bij bepaalde waarden van het Y-register dingen die volgens de definitie van OSRDCH absoluut niet mogen gebeuren. Ook worden eventueel ingetypte control codes voorzien van een echo via OSWRCH en ook dit mag niet volgens de definitie van OSRDCH. Kortom er wordt onverantwoord omgesprongen met een belangrijke systeemroutine als OSRDCH en dat is een zeer kwalijke zaak.

UITBREIDING JVDU-SOFT

Om kort te zijn vind ik dat het merendeel van de vrije control codes bestemd zou moeten worden voor algemene zaken waar elke VDU wat aan heeft. Daarnaast moet er ruimte blijven (of gecre-eerd worden) om de specifieke mogelijkheden van de diverse CRTC's te kunnen benutten. Ik heb een relatief simpel en elegant concept bedacht waarbinnen dit alles via het OSWRCH kanaal mogelijk is. Het houdt onder meer in dat de control code ruimte uitgebreid wordt tot in principe 256 stuks. Daarbij biedt het concept de mogelijkheid om control codes exclusief naar een bepaalde driver te sturen, d.w.z. alleen naar de beeldschermdriver en niet ook naar de printerdriver en omgekeerd. Dit concept is opwaarts compatible met de originele ATOM soft. Alle oude software blijft dus onveranderd bruikbaar.

CODE AFSPRAKEN

Om zo'n concept daadwerkelijk van de grond te krijgen, moeten zeer duidelijke afspraken gemaakt worden over welke codes voor wat gebruikt gaan worden, hoe ze in de VDU-soft ge-implementeerd moeten worden en hoe ze in de praktijk gebruikt moeten worden. Ook zullen enkele extra RAM locaties als werkruimte aan de VDU soft toegewezen moeten worden.

Het grootste struikelblok is echter dat een eventuele JVDU uitgebreid volgens dat nieuwe concept NIET compatible zal zijn met de huidige VDU-soft's (2.8 en 3.3) voor wat betreft het gebruik van nieuwe control codes. Dat zal eveneens betekenen dat de reeds voor VDU-2.8 en/of 3.3 geschreven software m.b.t. die nieuwe control codes herschreven moet gaan worden. Ik besef terdege dat dit voor velen absoluut onacceptabel is. Ik heb met mijn concept wat dat betreft wellicht te hoog gegrepen in mijn nobel streven naar optimale uitwisselbaarheid van software.

In het verleden is overigens al eens een poging gedaan om tot een min of meer standaard opzet van 80 koloms soft te komen. Wat dat betreft verwijs ik naar een artikel van B.Poot in AN nr.4 jaarg.5 blz. 46/47. Ik heb mij voorlopig beperkt tot tot deze "no nonsense" JVDU 0.0 soft. Gewoon 80 (of 40) kolommen en verder niets. Mocht het tot een vervolg komen, dan zal ik dat niet "op eigen houtje" doen, maar in goed overleg met anderen.

J. Jobse,
Duinweg 35,
4356 AP Oostkapelle, (regio Zeeland)
Tel. 01188 - 1987.

```
10  REM ***** JVDU-0.0 *****
20
30  REM J.Jobse. febr 1989.
40
50  REM Dit is een minimale 80/40
60  REM vdu-soft voor EF9345 CRTC.
70
80  REM #00..#1F standaard ATOM ctrl's
90  REM #20..#7F printbare ASCII kar's
100 REM #80..#FF semi-graphic mosaics
110
120 REM LET OP: codes #80..#FF kunnen
130 REM in latere versies eventueel
140 REM een andere betekenis krijgen.
150 REM Assembleren met SALFAA 2.5.
160
170 IN."SYMBOLTABLE ADR (1.5 K) "T
180 IN."BEGIN-ADRES CODE "C
190 IN."OPSLAG-ADRES (3/4 K) "R
200 P."PASS 1"';PASS 0;GOSUB 270
210 P."PASS 2"';PASS 1;GOSUB 270
220 P."JVDU0.0"&start80,&vdu'end,'
230 P."START80"&start80,'
240 P."START40"&start40,'
250 END
260
270 ASM-BEGIN
280
290         .OPTION :01000000
300         .TABLE T,T+#600
310         .CODE C
320         .RAM F
330
340          \ flags:   b7 = scr'flg
350          \          b0 = esc'flg
360          \
370          \ curpos: b5..b0 = pos'cntr
380          \
390          \ cursor: b7 = on/off
400          \          b6 = flash
410          \          b5 = underscore
420          \
430          \ mode :  b7 = 80/40 col'mode
440          \          b6 = page'mode
450          \          b4..b0 = page'lines
460
470 :buf'beg= #52
480 :key    = #DE
490 :flags  = #DF
500 :curpos = #E0
510 :cursor = #E1
520 :tmp    = #E2 \ 2
530 :xsave  = #E4
540 :ysave  = #E5
550 :ende   = #E6
560 :lmask  = #E7
570
580 :stack = #0100
590 :wrcvec= #0208 \ 2
600 :rdcvec= #020A \ 2
610 :ppia   = #B000
620 :portA  = ppia+0
630 :portB  = ppia+1
640 :portC  = ppia+2
650 :crtc   = #BE20 \ 16
660 :status= crtc+0 \ R
670 :cmd    = crtc+0 \ W
680 :bytC   = crtc+1
690 :bytB   = crtc+2
700 :bytA   = crtc+3
710 :ayptr  = crtc+4
720 :axptr  = crtc+5
730 :myptr  = crtc+6
740 :mxptr  = crtc+7
750
760 :wait       = #FB83
770 :beep       = #FD1A
780 :frame      = #FE66
790 :scan'kb    = #FE71
800 :to'printer= #FEFB
810 :osrch      = #FFF4
820
830 :econ= #06
840 :bell= #07
850 :bs   = #08
860 :ht   = #09
870 :lf   = #0A
880 :vt   = #0B
890 :cls = #0C
900 :cr   = #0D
910 :pgon= #0E
920 :pgof= #0F
930 :scof= #15
940 :can = #18
950 :esc = #1B
960 :home= #1E
970 :del = #7F
980
990 :skip2 = #2C
1000
1010 :curs= #40 \ vast blokje
1020 :fgnd= #07 \ wit
1030 :sexel= #08
1040 :krf = #00
1050 :krl = #50
1060 :nop = #91
1070 :iny = #80
1080 :ind = #80
1090 :wr  = #00
1100 :rd  = #08
1110 :inc = #01
1120
1130 :tgs = 1
1140 :mat = 2
1150 :pat = 3
1160 :dor = 4
1170 :ror = 7
1180
```

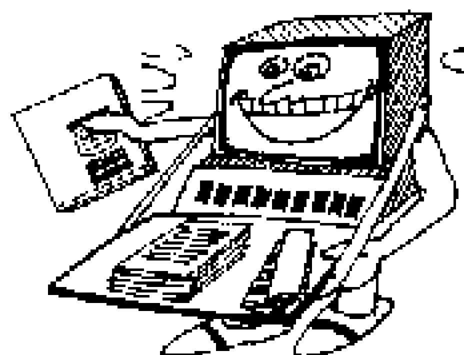
```
1190            \ VDU ROUTINES
1200
1210 :start80 SEC
1230          BCS skp'clc
1240 :start40 CLC
1260 :skp'clc LDY @3
1280 :set'vec LDA vec'tbl,Y
1300          STA wrcvec,Y
1310          DEY
1320          BPL set'vec
1330 :init'crtc
1340          LDA @#00
1350          ROR A
1360          STA mode
1370          LDA @nop
1380          JSR exec'cmd
1390          LDX @ror
1400          LDA @#08
1410          JSR wr'reg
1420          LDX @dor
1430 :nxt'reg LDA reg'inh40-1,X
1450          BIT mode
1460          BPL skp'inh80
1470          LDA reg'inh80-1,X
1480 :skp'inh80
1490          JSR wr'reg
1500          DEX
1510          BNE nxt'reg
1520          STX flags
1530          STX bytB
1540          BIT mode
1550          BMI stor'bytA
1560          LDX @16*fgnd
1570 :stor'bytA
1580          STX bytA
1590          LDA @2*curs
1600          STA cursor
1610
1620 :cls'    JSR disable'curs
1640          JSR home'
1650          STA portA \ A=3
1660          LDX @24
1670 :nxt'clr JSR clr'line
1690          JSR inc'myptr
1700          DEX
1710          BNE nxt'clr
1720          JMP enable'curs
1730
1740 :home'   JSR reset'page
1760          JSR rd'ror
1770          STA myptr
1780
1790 :cr'     LDA @0
1810          STA mxptr
1820 :hw \ not used ctrl code
1830 :can'    RTS
1850
1860 :reset'page
1870          BIT mode
1880          BVC rts1
1890 :pgon'   LDA mode
1910          AND @#80
1920          ORA @#40+23
1930          BNE to'mode
1940
1950 :pgoff   LDA mode
1970          AND @#80
1980 :to'mode STA mode
2000          RTS
2010
2020 :ht'     LDA @krl+rd+inc
2040          BIT mode
2050          BMI do'ht
2060 :ht40    LDA @krf+rd+inc
2080 :do'ht   JSR exec'cmd
2100          JMP tstx
2110
2120 :bs'     LDA mxptr
2140          BNE no'wrap
2150 :wrap    JSR rd'ror
2170          CMP myptr
2180          BEQ rts1
2190          LDA mode
2200          AND @#80
2210          ORA @39
2220          STA mxptr
2230
2240 :vt'     JSR rd'ror
2260          CMP myptr
2270          BEQ rts1
2280 :dec'myptr
2290          LDY myptr
2300          DEY
2310          CPY @8
2320          BGE skp'mod24
2330          LDY @8+23
2340 :skp'mod24
2350          STY myptr
2360 :rts1    RTS
2380
2390 :no'wrap BIT mode
2410          BPL skp'eor
2420          EOR @#80
2430          STA mxptr
2440          BPL skp'dec
2450 :skp'eor DEC mxptr
2470 :skp'dec RTS
2490
2500 :delete  JSR bs'
2520          LDA @" "
2530
2540 :print'ch
2550          JSR transf'ch
2560 :tstx    LDA mxptr
2580          BNE rts0
2590
2600 :lf'     JSR disable'curs
2620          BIT mode
```

```
2630            BVC go'on
2640            LDA mode
2650            AND @#1F
2660            BNE go'on'
2670 :key'wait
2680            JSR scan'kb
2690            BCS key'wait
2700            TYA \ space?
2710            BNE go'on
2720            JSR pgon'
2730            INC mode
2740 :go'on'    DEC mode
2760 :go'on     JSR inc'myptr
2780            JSR rd'ror
2790            CMP myptr
2800            BNE no'scroll
2810 :scroll    ADC @0 \ C=1
2830            CMP @8+24
2840            BLT skp'wrap
2850            LDA @8
2860 :skp'wrap
2870            JSR wr'reg
2880            LDX mxptr
2890            JSR cr'
2900            JSR clr'line
2910            STX mxptr
2920 :no'scroll
2930
2940 :enable'curs
2950            LDA cursor
2960 :to'mat    LSR A
2980 :wr'mat    LDX @mat
3000
3010 :wr'reg    STA bytC
3030            TXA
3040            ORA @ind+wr
3050            BNE exec'cmd
3060
3070 :rd'ror    LDX @ror
3090
3100 :rd'reg    TXA
3120            ORA @ind+rd
3130            BNE exec'cmd
3140
3150 :transf'ch
3160            STA bytC
3170            LDA @krl+wr+inc
3180            BIT mode
3190            BMI exec'cmd
3200            LDA @krf+wr+inc
3210            .BYT skip2
3220
3230 :inc'myptr
3240            LDA @iny
3250
3260 :exec'cmd
3270            STA cmd+exec
3280 :busy      BIT status
3300            BMI busy
3310            LDA bytC
3320 :rts0      RTS
3340
3350 :disable'curs
3360            LDA cursor
3370            AND @#60
3380            BPL to'mat
3390
3400 :clr'line
3410            LDA @" "
3420 :clr'ln    JSR transf'ch
3440            BVC clr'ln
3450 :rts2      RTS
3470
3480 :esc'      LDA portA
3500            AND @#F0
3510            BEQ rts2
3520            LDA flags
3530            LSR A
3540            BCS scr'on
3550            INC flags
3560            RTS
3570
3580 :scr'on    LDA @#00
3600            STA portA
3610            LDA flags
3620            AND @#7E \ clr b7+b0
3630 :to'flags
3640            STA flags
3650            RTS
3660
3670 :scr'off   LDA flags
3690            ORA @#80
3700            BNE to'flags
3710
3720            \ SCHRIJF ROUTINES
3730
3740 :wr'char   JSR to'printer
3760            PHP
3770            PHA
3780            CLD
3790            STY ysave
3800            STX xsave
3810            JSR to'screen
3820            JMP #FE5F
3830
3840 :to'screen
3850            CMP @scon
3860            BEQ scr'on
3870            CMP @scof
3880            BEQ scr'off
3890            BIT flags
3900            BMI rts2
3910            PHA
3920            LDA curpos
3930            BIT mode
3940            BPL to'exp
3950            LSR A
3960            BCC to'exp
```

```
3970            ORA @#80
3980 :to'mxp    STA mxptr
4000            PLA
4010            CMP @del
4020            BEQ is'del
4030            CMP @" "
4040            BLT is'ctrl
4050            BIT mode
4060            BMI print'it
4070            TAY \ <#80 ?
4080            BPL print'it
4090            LDY @#20 \ graph40
4100            STY bytB
4110 :print'it
4120            JSR print'ch
4130 :restore   LDA @0
4150            STA bytB
4160            LDA mxptr
4170            BIT mode
4180            BPL skp'transl
4190            ASL A
4200            ADC #0
4210 :skp'transl
4220            STA curpos
4230            RTS
4240
4250 :is'del    JSR delete
4270            LDA @bs
4280
4290 :is'ctrl   SEC
4310            SBC @7
4320            BCC restore
4330            TAX
4340            LDA ctrl'l,X
4350            STA tmp
4360            LDA ctrl'h,X
4370            STA tmp+1
4380            JSR #FEC1
4390            JMP restore
4400
4410            \ LEES ROUTINES
4420
4430 :ctrl'echo
4440            JSR oswrch
4450
4460 :rd'char  JSR rd'char'
4480            PHA
4490            TSX
4500            LDA stack+3,X     \ !!!!!!!!
4510            CMP @>#CD4A-1     \ test of
4520            BEQ is'buf'call   \ aanroep
4530            CMP @>#FFE9-1     \ uit input
4540            BNE no'buf        \ routine
4550            LDA stack+5,X     \ (#CD0F)
4560            CMP @>#CD1B-1     \ komt
4570            BEQ is'buf'call   \ !!!!!!!!
4580 :no'buf    LDX xsave
4600            PLA
4610            RTS
```

```
4620
4630 :is'buf'call
4640            LDX xsave
4650            JSR reset'page
4660            PLA
4670            BIT buf'beg \ #00/#40
4680            CMP @" "
4690            BGE is'char
4700            CMP @esc
4710            BEQ ready
4720            CMP @cr
4730            BEQ ready
4740            CMP @can
4750            BNE ctrl'echo
4760 :is'cancel
4770            LDY @#00
4780            BVC to'beg
4790            LDY @#40
4800 :to'beg    STY buf'beg
4820            RTS
4830
4840 :is'char   CPY @#00+63
4860            BVC skp'cpy
4870            CPY @#40+63
4880 :skp'cpy   BLT ready
4900            BEQ no'del'cor
4910            CMP @del
4920            BEQ del'corr
4930            CPY @#00+125
4940            BVC skp'cpy'
4950            CPY @#40+125
4960 :skp'cpy'
4970            BGE ready
4980 :no'del'cor
4990            CMP @del
5000            BEQ ready
5010            INC buf'beg
5020 :ready     RTS
5040
5050 :del'corr
5060            TYA
5070            SBC @64 \ C=1
5080            STA buf'beg
5090            LDA @del
5100            RTS
5110
5120 :rd'char'
5130            PHP
5140            CLD
5150            STX xsave
5160            STY ysave
5170 :nxt'rd    JSR read'key
5190            CPY @5
5200            BEQ is'lock
5210            BCC no'curs
5220            CPY @8
5230            BCC is'curs
5240            CPY @14
5250            BEQ is'copy
```

```
5260 :no'curs JMP #FEB1
5280
5290 :is'lock LDA lmask
5310          EOR @#60
5320          STA lmask
5330          BCS nxt'rd
5340
5350 :is'curs TYA
5370          AND @#05
5380          ROL portB
5390          ROL A
5400          JSR to'screen
5410          JMP nxt'rd
5420
5430 :is'copy LDA @kr1+rd
5450          BIT mode
5460          BMI copy80
5470 :copy40  LDA @krf+rd
5490 :copy80  JSR exec'cmd
5510          JMP #FEAD
5520
5530 :read'key
5540          LDA @16 \ 1/3 sec
5550 :nxt'scan
5560          LDX @2 \ snel rept
5570          BIT portC \ REPT-key
5580          BVC repeat
5590          JSR frame
5600          PHA
5610          JSR scan'kb
5620          PLA
5630          LDX @4 \ debounce
5640          BCS repeat
5650          DEX \ slow rept
5660          CPY key
5670          BEQ repeat
5680          SEC
5690          SBC @1
5700          BNE nxt'scan
5710 :repeat  JSR wait
5730 :no'key' STY key
5750          JSR scan'kb
5760          BCS no'key'
5770          JSR scan'kb
5780          BCS no'key'
5790          RTS
5800
5810          \ TABELLEN
5820
5830 :vec'tbl .WRD wr'char,rd'char
5850
5860 :reg'inh40
5870         .BYT #10,#50,#6E,#00
5880
5890 :reg'inh80
5900         .BYT #D0,#50,#6E,#00+fgnd
5910
5920 :ctrl'l .BYT beep,bs',ht'
5940         .BYT lf',vt',cls'
5950         .BYT cr',pgon',pgoff
5960         .BYT nu,nu,nu
5970         .BYT nu,nu,scr'off
5980         .BYT nu,nu,can'
5990         .BYT nu,nu,esc'
6000         .BYT nu,nu,home'
6010         .BYT nu
6020
6030 :ctrl'h .BYT >beep,>bs',>ht'
6050         .BYT >lf',>vt',>cls'
6060         .BYT >cr',>pgon',>pgoff
6070         .BYT >nu,>nu,>nu
6080         .BYT >nu,>nu,>scr'off
6090         .BYT >nu,>nu,>can'
6100         .BYT >nu,>nu,>esc'
6110         .BYT >nu,>nu,>home'
6120         .BYT >nu
6130
6140 :vdu'end
6150
6160          .END
6170  RETURN
```



```
**************************************************************
*   ZOALS HET KLOKJE THUIS TIKT, TIKT HET NERGENS            *
**************************************************************
```

De klok gerepareerd door Roland Leurs.

In Atomnieuws 5.4 staat het programma 'KLOK'. Dit programma geeft samen met de bijbehorende datafile 'FILMNEG' een mooie grote digitale klok op het scherm. Maar wat blijkt nu ... de 9 doet het niet. Wat jammer, klok kapot ....

In de listing staat nog mooi vermeld:

```
   10 REM (C) MICROBE 1985
```

dus zo weten wij meteen welk genie dit programma gemaakt heeft, maar toen ik in het telefoonboek meneer Microbe wilde opzoeken bleek dat er geen meneer Microbe in staat. Zelfs 008 kon mij niet helpen. Dus moest ik de gereedschapskist maar onder het stof uithalen en zelf gaan sleutelen.

Ik zelf zit op een HTS dus een digitale klok repareren moet eigenlijk best meevallen en wat bleek nu?

Meneer Microbe had het laatste stuk van FILMNEG vervangen door een stuk uit een adressenbestand. Ik heb met een team van 1 Atom, 1 scherm en 2 drives geprobeerd om een negen te maken en dat is gelukt. Alles wat U moet doen is de regioschijf uit de kast halen met FILMNEG op (schijfke 4b van 1986) en laadt de datafile. Voer nu even onderstaande data in :

```
7E00 F0 00 00 0F F1 00 00 1F 1F 00 00 F1 FF FF FF FF
7E10 FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0F FF
7E20 FF 80 01 FF FF 00 00 FF FC 00 00 3F F8 00 80 1F
7E30 F0 03 E0 0F E0 07 F0 07 E0 07 F0 07 C0 0F F8 03
7E40 C0 0F F8 03 C0 0F F8 03 C0 0F F8 01 C0 0F F8 01
7E50 C0 0F F8 01 C0 0F F8 01 C0 0F F8 00 E0 07 F0 00
7E60 E0 07 F0 00 F0 03 E0 00 F0 00 80 00 F8 00 00 00
7E70 FC 00 00 00 FF 80 20 00 FF F0 E0 00 FF FF E0 00
7E80 FF 0F E0 00 FE 0F E0 01 F8 07 E0 01 F0 03 E0 01
7E90 E0 03 E0 01 E0 03 E0 03 C0 03 C0 03 C0 07 C0 03
7EA0 E0 0F 80 07 E0 02 00 07 F0 00 00 0F F0 00 00 1F
7EB0 F8 00 00 3F FE 00 00 FF FF 80 01 FF FF F0 0F FF
7EC0 FF 80 01 FF FF 00 00 FF FC 00 00 3F F8 00 80 1F
7ED0 F0 03 E0 0F E0 07 F0 07 E0 07 F0 07 C0 0F F8 03
7EE0 C0 0F F8 03 C0 0F F8 03 C0 0F F8 01 C0 0F F8 01
7EF0 C0 0F F8 01 C0 0F F8 01 C0 0F F8 00 E0 07 F0 00
7F00 E0 07 F0 00 F0 03 E0 00 F0 00 80 00 F8 00 00 00
7F10 FC 00 00 00 FF 80 20 00 FF F0 E0 00 FF FF E0 00
7F20 FF 0F E0 00 FE 0F E0 01 F8 07 E0 01 F0 03 E0 01
7F30 E0 03 E0 01 E0 03 E0 03 C0 03 C0 03 C0 07 C0 03
7F40 E0 0F 80 07 E0 02 00 07 F0 00 00 0F F0 00 00 1F
```

```
7F50 F8 00 00 3F FE 00 00 FF FF 80 01 FF FF F0 0F FF
7F60 FF 80 01 FF FF 00 00 FF FC 00 00 3F F8 00 80 1F
7F70 F0 03 E0 0F E0 07 F0 07 E0 07 F0 07 C0 0F F8 03
7F80 C0 0F F8 03 C0 0F F8 03 C0 0F F8 01 C0 0F F8 01
7F90 C0 0F F8 01 C0 0F F8 01 C0 0F F8 00 E0 07 F0 00
7FA0 E0 07 F0 00 F0 03 E0 00 F0 00 80 00 F8 00 00 00
7FB0 FC 00 00 00 FF 80 20 00 FF F0 E0 00 FF FF E0 00
7FC0 FF 0F E0 00 FE 0F E0 01 F8 07 E0 01 F0 03 E0 01
7FD0 E0 03 E0 01 E0 03 E0 03 C0 03 C0 03 C0 07 C0 03
7FE0 E0 0F 80 07 E0 02 00 07 F0 00 00 0F F0 00 00 1F
7FF0 F8 00 00 3F FE 00 00 FF FF 80 01 FF FF F0 0F FF
```

Save nu de file FILMNEG opnieuw met
*SAVE "FILMNEG" 3F00 8000
en dan kunt U ook genieten van een mooie 9 als het 19:59:59 is !

Succes ermee (vooral met intikken) en ....

.... de vriendelijke groeten van Roland !

NOGMAALS CHECKER

door Roland Leurs

Er is helaas iets fout gegaan bij de verspreiding van CHECKER opregioschijf ANH88-12. Het programma op de schijf bevatte ook de CRC waarde van iedere regel, zoals was afgedrukt in het bijbehorende boekje. Dat geeft tijdens het runnen al meteen een foutmelding. Zulke dingen gebeuren als je geen programma meestuurt naar de redactie.

Op de nieuwe schijf staat dus het programma CHECKER zoals het dus wel loopt. Mijn excuses voor het bezorgde ongemak.

Met de vriendelijke groeten van Roland.

SERIELE- PLUS PARALLELLE-UITGANGS KAART.

Dat de ATOM een pracht van een computer is, mag wel als bekend worden verondersteld. Toch heeft hij nadelen. Een daarvan is dat hij geen seriele uitgang heeft. Daar is echter iets aan te doen.
In de familie van de 6500 serie zit een ic die speciaal voor dit doel is gemaakt, namelijk de 6551. Parallelle data, die aan dit ic wordt aangeboden, komt er serieel inclusief start- en stopbits en pariteit-sbits weer uit. Dit uitgangssignaal kan als V24 (RS 232) signaal gebruikt worden. Andersom, serieel aangeboden signalen, kunnen naar parallel worden omgezet, zodat de ATOM ze weer begrijpt. De spanningsniveaus zijn voor de uitgang +5V-0V en voor de ingang mag dit +12V en -12V zijn.

Omdat er nog ruimte genoeg op een euro-kaartje overbleef, is er ook nog een parallelpoort bij gemaakt. Hiervoor is het (zeer) goedkope ic 6821 gebruikt. (Fl. 2,50 bij de Elektronica dump in Eindhoven.) Dit ic lijkt wat op de bekende I/O 6522, doch heeft geen timers.

Als overige onderdelen treffen we nog twee 74LS-ers als adresdecodering, een 74240 als buffer, een kristal voor het opwekken van de juiste frequenties voor de seriele dataoverdracht, een paar ontkoppel condensatortjes en een clamping cirquitje aan. Omdat de IC-'s die gebruikt worden zo ingewikkeld zijn, kan de schakeling simpel blijven.

DE BOUW:

Om een print te maken, is er een lay-out beschikbaar op transparant. Het grootste probleem is hiermee dus al opgelost. Na het etsen en boren van de enkelzijdige print worden de, op het figuur 1 aangegeven, draadbrugjes ,de dioden, weerstandjes, condensators en kristal gesoldeerd. Een van de twee benodigde 64 polige AD-connectoren heeft nog een voorbewerking nodig. De hieronder genoemde pennen (van een connector dus) dienen verwijderd te worden:

pen 1a 1b
pen 2a 2b
pen 16a 16b
pen 17a 17b
pen 31a 31b
pen 32a 32b

Nu is deze connector geschikt voor twee flacable aansluitingen. Twee connectoren van een bijbehorend type voor printmontage passen niet op de korte zijde van een euro kaart, dus daarom heb ik voor deze oplossing gekozen.
Als de connectoren en de ic's geplaatst zijn, is de kaart bijna klaar.
We moeten nu nog bepalen op welke adressen de poorten te bereiken zijn.
Er is een grove instelling, met de keuze van het B4 of BC blok.(zie fig 2) en een fijnere per poort voor Bx00, Bx40, Bx80 en BxC0 als basisadres.(fig 3)

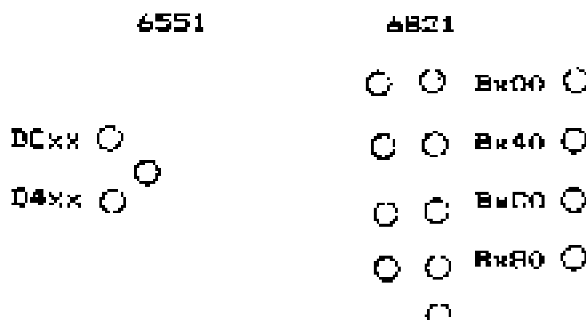


fig. 2 fig 3

Als de keuze is bepaald en ook deze draadbrugjes zijn geplaatst, is de print klaar.

IN NAAM VAN ATOM, DOE OPEN DE POORT.

Nee, geen vaderlandse geschiedenis, maar structurele probleempjes van de ATOM. De adressen die de kaart nodig heeft, zijn namelijk bedoeld voor intern gebruik, oftewel de busbuffer ic4 in de ATOM blijft dicht. De makkelijkste oplossing om dit te verhelpen is het ic4 verwijderen en de datalijnen doorverbinden. Andere oplossingen moeten individueel gevonden worden, i.v.m. de indeling van de memorymap binnen en buiten de ATOM.

HET PROGRAMMEREN.

Om de twee poorten te laten werken, zoals de gebruiker dat wil, moet je ze vertellen wat van ze verwacht wordt. Net als bij b.v. de 6522 moeten ook deze ic's geinitialiseerd worden. We zullen beide eens nader bekijken.

De 6551:
Er zitten zes registers in dit ic, die bereikt kunnen worden door de twee adressen (RS0 en RS1) in combinatie met het write of read signaal 1 of 0 te maken. Tabel 1 geeft hiervan een overzicht.

RS1	RS0	Write	read
0	0	transmit data	receive data
0	1	softw. reset	status
1	0	command register	
1	1	control register	

Indien men schrijft in adres #Bxx1 worden de registers softwarematig gereset. De data waarmee men schrijft is niet van belang. Bij de volgende tabellen wordt tevens aangegeven hoe de inhoud van de registers zijn na zo'n softwarereset en na een hardwarereset (break of na inschakelen).

Het statusregister
Hieruit kan men de toestand lezen.

7 6 5 4 3 2 1 0

0= geen pariteitsfout
1= pariteitsfout

0= geen formaatfout
1= formaatfout

0= geen overrun
1= overrun

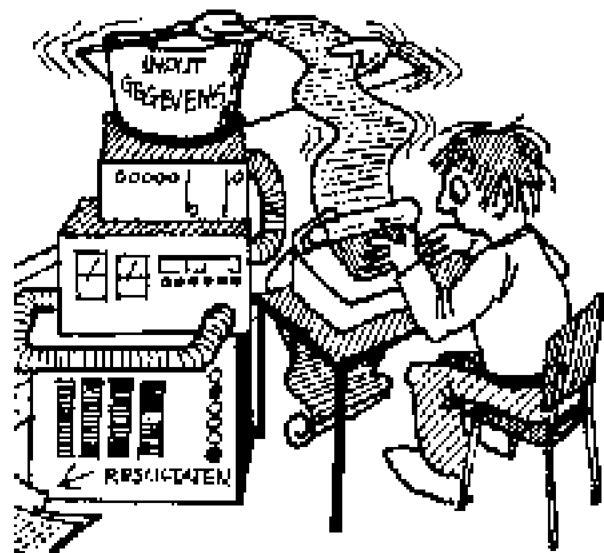
0= receive data register niet vol
1= receive dataregister vol

0= transmit data register niet leeg
1= transmit data register leeg

0= data carier detect gezien
1= data carier detect niet gezien

0= data set klaar
1= data set niet klaar

0= geen interrupt gekregen
1= interrupt gekregen



hardware reset 0 - - 1 0 0 0 0
software reset - - - - - 0 - -

Het commando register.
Stelt o.a. de vorm van het bericht in.
7 6 5 4 3 2 1 0

0= ontvanger en interrupts uit
1= ontvanger en interrupts aan

0= interrupt via bit 3 van statusreg. toegestaan.
1= interrupt niet toegestaan.

0 0 = transmit interr. uit, RTS hoog, zender uit.
0 1 = transmit interr. aan, RTS laag, zender aan.
1 0 = transmit interr. uit, RTS laag, zender aan.
1 1 = transmit interr. uit, RTS laag, zender break.

0= normaal
1= echo, bit 2 en 3 moeten 0 zijn.

- - 0 = geen pariteitsbit
0 0 1 = oneven pariteitbit
0 1 1 = even pariteitsbit
1 0 1 = mark pariteitsbit zenden,
        geen controle
1 1 1 = space pariteitsbit zenden,
        geen controle

hardware reset 0 0 0 0 0 0 0 0
software reset - - - 0 0 0 0 0

Het control register.
Hiermee wordt o.a. de zendsnelheid gezet.

7 6 5 4 3 2 1 0

0 0 0 0 = externe klok.
          (niet gebruikt).
0 0 0 1 = 50 baud
0 0 1 0 = 75 baud
0 0 1 1 = 109,92 baud
0 1 0 0 134,58 baud
0 1 0 1 = 150 baud
0 1 1 0 = 300 baud
0 1 1 1 = 600 baud
1 0 0 0 = 1200 baud
1 0 0 1 1800 baud
1 0 1 0 = 2400 baud
1 0 1 1 = 3600 baud
1 1 0 0 = 4800 baud
1 1 0 1 = 7200 baud
1 1 1 0 = 9600 baud
1 1 1 1 = 19200 baud

0= externe klok voor ontvangen
1= baudrate generator

0 0 = woordlengte 8 bits
0 1 = woordlengte 7 bits
1 0 = woordlengte 6 bits
1 1 = woordlengte 5 bits

0= 1 stopbit
1= 2 stopbits, 1 stopbit als woord-
      lengte = 8 bits + parity, 1 1/2
      stopbits als woordlengte = 5
      bits + geen parity.

hardware reset 0 0 0 0 0 0 0 0
software reset - - - - - - - -

De transmit en receive registers.
Deze spreken voor zich. Hier komt dus de te verzenden, of de ontvangen code in te staan.

Ik kan me voorstellen, dat nog niet voor een ieder alles duidelijk is. Daarom staat er achter dit artikeltje een simpel voorbeeld-programmaatje.

De 6821.
Zoals al eerder is gezegd, lijkt de 6821 nogal op de 6522. Hij heeft b.v. ook twee 8-bits poorten, die per bit afzonderlijk als in of uitgang geinitialiseerd kunnen worden. Er zijn zes registers, die met de adres lijnen RS1 en RS0, tezamen met een bitje uit de controleregisters geselecteerd kunnen worden.
Adressering van de registers.

RS1	RS0	CRA-2	CRA-1	
0	1	-	-	control reg. A
0	0	0	-	data dir.reg. A
0	0	1	-	periferiereg. A
1	1	-	-	control reg. B
1	0	-	0	data dir.reg. B
1	0	-	1	periferiereg. B

Het controle register.

De controle registers van poort A en B zien er het zelfde uit.

7 6 5 4 3 2 1 0

zetten de aparte interupt lijnen CA1 in de juiste mode.

0= data directin register selectie.
1= periferie register selectie

zetten de aparte interupt lijnen CA2 in de juiste mode.

interuptvlag van CA2

interuptvlag van CA1

Het data direction register.

Het data direction register geeft aan of een bitje gezien moet worden als uit- of ingang. Selecteer het data direction-register en zet hierin een 1 voor uitgang en 0 voor ingang.

Voorbeeld;

```
?#Bxx1 = 0      selecteer het data dir.reg.
?#BxX0 = #0F    van poort A is bit 0 t/m3 uitgang en
                4 t/m 7 ingang.
?#Dxx1 = 4      Selecteer nu het perifeerieregister,
                en op #Bxx0 kan met de uit/ingang ge-
                werkt worden.
```

VOORBEELD PROGRAMMATJE.

Type het onderstaand programmaatje in en men kan de seriele interface proberen. Gebruik hierbij een connector, waarvan de punten 2 en 3 aan elkaar liggen. Ook de punten 4, 5, 6 en 8 moeten gekoppeld zijn. Let tevens op de adressering van de kaart. In dit voorbeeld is het basisadres van de 6551 #BC40. Als alles goed gaat, verschijnt er een tekst op het scherm.
Veel plezier verder met deze kaart.

```
10P.$12"IO 6821 TEST"''
15 IN."WAT IS HET BEGINADRES?"X
20?(X+1)=0;       REM SEL DDR
30?X=#FF;         REM OUTPUT
40?(X+1)=4;       REM SEL. PER
50?(X+3)=0;       REM SEL DDR
60?(X+2)=#FF;     REM OUTPUT
70?(X+3)=4;       REM SEL. PER
80
90?X=#FF
100?(X+2)=#FF
110PAUSE 20
120?X=0
130?(X+2)=0
140PAUSE 20
150G.90
```

```
  1 P.$12
 10 IN."BASIS-ADRES RS232 INTERFACE"A
 20 RESTORE
 30 ?(A+1)=0
 40 ?(A+3)=#16
 50 ?(A+2)=#0B
 60 DO
 70 READ X
 75 WAIT
 80 ?A=X
100 P.$?A
110 UNTIL X=62
120 G.20
130 DATA 65,84,79,77,32,82,83,50,51,50,32,84,69,83,84,32,79,46
140 DATA 75,46,10,13,62
```

fig. 3

8BH1.3

fig.2

BAUD.GNR.

fig.3

74LS240

6850

74LS138

Z80

fig.2

4X10K

8BH15

74LS138

LET OP : NIET EXAKT OP JUISTE SCHAAL

EDOS-GDOS AANPASSINGEN EN PROBLEMEN

Bij het opstellen van de tekst over EDOS die geplubliceerd werd in nummer 4 van jaargang 7 zijn enkele kleine foutjes geslopen.

1) Het 3de alinea op pag. 14 moet zijn:
"Willen we nu in een machinetaal programma een bepaald aantal
"sectoren van een disk inlezen in het geheugen dan hebben we alleen
"maar de volgende bytes in te vullen.
Verder is deze alinea korrekt.

2) Het laatste alinea op pag. 19 handelend over de RAMDSKL moet zijn:
"Deze ramdisk is in rom 5. De derde driver in deze rom en de data
"start van #7E00 naar #0000 toe; d.w.z. dat de directory zich
"bevindt in de blokken #7F00 voor de file namen en #7E00 voor de
"file pointers.

Bij het gebruik van de Gdos kaart met de Edos software kwamen de volgende problemen aan het licht.

1) De reset voor het controller Ic WD1793 word geleverd enerzijds door de normale reset puls, anderzijds door de VIA (6821 pin 6).
Dit heeft tot gevolg dat indien in de drive tabel het option byte niet goed geconfigureed is, het disk controller Ic steeds gereset blijft zodat de controller nooit werkt.
De correcte waarde voor het option byte is zodanig dat bit 4 steeds gezet is, dus #10.
Indien de Gdos-kaart enkel gebruikt word voor Edos kan men beter deze baan doorsnijden vlak tegen pin 6 van de VIA (6821), de diode verwijderen en in de plaats van deze diode een draadbrugje plaatsen.

2) Een volgend probleem kwam aan het licht bij het gebruik van bepaalde drives zoals deze van MPI. Deze deselecteren zichzelf niet wanneer de motor stopt.
Dit heeft tot gevolg dat het select lampje blijft branden, maar wat nog erger is, is dat hierdoor de stepper motor bloedheet loopt, wat overmatig verwarming van de drive tot gevolg heeft.
Om dit euvel op te lossen kan men beter volgende verbetering aanbrengen.
Neem Ic14 (74LS06) uit het voetje, plooi de pennetjes 5-6-8-9 naar boven plaats de Ic terug in het voetje, zodanig dat de pennetjes 5-6-8-9 niet meer in het voetje zitten.
Maak nu een draadverbinding tussen pen 3 van Ic8 (743LS8) en pen 10 van P12. Maak ook nog een draadverbinding tussen pen 11 van Ic8 (74LS08) en pen 1 van P12 (dit zijn de drive select lijnen).
Zie bijgeleverd schema "SELECT TURN OFF ON MOTOR OFF".

3) Sommige drives hebben geen READY lijn. Dit heeft tot gevolg dat de software de drive nooit ready ziet en niets wil doen. Dit kan ook voorkomen worden door tussen pen 4 van Ic16 (74LS153) en de aardpen (8) een draadverbindisng aan te brengen, beter is nog dit Ic16 volledig te verwijderen indien U geen PHILIPS drives

gebruikt maar standaart SUGARD compatible drives, U verbindt dan met een stukje draad de pennen 7 en 4 van dit Ic met elkaar, of indien U helemaal geen READY signaal van de drive krijgt verbindt U pen 7 met pen 8 (aarde).
Indien U twee verschillende drives heeft, d.w.z. een met ready en een zonder, gebruik dan het READY systeem niet, verbindt het zoals in voorgaande is uitgelegd.

4) Verder waren er nog problemen met de dubbelstap schakeling. Deze problemen deden zich vooral voor in het zoeken naar track 0, waardoor de kop soms tussen twee tracks belandde .
Dit werd opgelost in de software en hierdoor kan men nu volstaan met een eenvoudiger schakeling dan de originele. Dit schema zal verderop nog getoond worden. Het principe is dat bij achterwaarts zoeken van een track eerst wordt gezocht naar track 0, hetgeen kan gebeuren in enkele stapmode, zodat track 0 steeds wordt gevonden.
Verder word dan het voorwaards zoeken uitgevoerd in dubbelle stap mode.
Zie ook het bijgeleverd schema "DUAL STEP PULSE CREATER".
De waarden in dit schema aangebracht zijn voor 6-3 ms stepping rate drives, indien uw drive trager is gelieve dan condencator C1 in evenredigheid aan te passen.

5) Daar het headload circuit gebruikt word als tijdsmeting voor het op snelheid komen van de drive, kan men beter C1 vergroten tot 30 µF, voor extra trage drives to 100 µF. Houdt hier wel rekening met het feit dat deze condensator een hoge doorslag spanning nodig heeft dit door de ompoling van het 74LS123 Ic (ongeveer 35V). Beter is het dit Ic te vervangen door een 74LS221. Deze heeft geen last van de ompoling, maar u moet dan ook condensator C2 verkleinen tot de helft 47 pF.

6) Er moet een verbinding gelegd worden tussen Ic4 pen 8 en Ic5 pen 32, indien het ready signaal van de drive gebruikt wordt anders ziet de software deze status niet.

7) De sector interleave voor 18 sectors/track met de DR1793 moet 2 zijn.Voor 16 sectors/track mag dit 1 zijn, en voor single density ATOM formaat moet dit 1 zijn.
Voor high density is de sectorinterleave 2.

8) De sector interleave byte voor de DR1793 driver was in de tabel aangegeven als N.U.. Dit is niet het geval en moet wel degelijk ingevuld worden. Zie verder in de nieuwe ontwikkelingen.

9) Indien een van volgende bytes 0 is: cylinders/schijf, trackinterleave, headinterleave of sectorinterleave dan kan men met deze drive niet formateren. Wel kan de drive gebruikt worden en zelfs geverifieerd indien het aantal cylinders/schijf is opgegeven.

NIEUWE ONTWIKKELINGEN

Nieuwe ontwikkelingen rond de Gdos-kaart zijn er ook gebeurd. Zo is

het nu mogelijk 1.2 Mega Byte drives van een IBM-AT machine aan te sluiten om aldus van deze grotere opslagcapaciteit gebruik te maken. Hier moet echter wel gewaarschuwd worden dat het Atoom dan ook sneller moet worden. Echt 2 Mhz hoeft niet, ikzelf run deze software op 1.5 Mhz met grote voldoening. Misschien werkt het ook op 1 Mhz (De timing van de drivers is zodanig geschreven dat zij bij 2 Mhz nog correct werken).

Hiervoor moet echter heel wat omgebouwd worden aan de Gdos kaart, o.a. moet het 8 Mhz Cristal vervangen worden door een 16 Mhz Cristal, daar deze drives hun data doorsturen met een snelheid van 500 Kbits/seconde. Ook moet de snelheid van de drive omgeschakeld worden van 300 toeren/minuut naar 360 toeren/minuut en tevens moet ook de schrijfstroom aangepast worden. Dit gebeurt normaal met pen 2 van P12. Indien deze hoog is, draait de drive aan hoge snelheid en is de schrijfstroom laag. wat nodig is voor een 1.2 Mbyte floppy. Ook moet op dit ogenblik de clock op 16 Mhz worden gebracht.

Dit alles kunnen we verwezelijken indien we pen 6 van de VIA vrij maken zoals hiervoor werd beschreven.
Verder hebben we nodig een output driverpoort zoals een 74LS06, een 74LS153 om de clock om te schakelen en een clockdeler 74LS393.
Aangezien dit alleen met standard drives (sugard aansluiting) kan werken is P12 in gebruik en P11 niet, en kunnen best de verbeteringen eerder beschreven uitgevoerd worden. Dit geeft 2 vrije poorten van het type 74LS06 en een vrij Ic van het type 74LS153.
Verder is nog een vrije counter aanwezig van het type 74LS393.
Dit alles kunnen we nu gebruiken om de hoge capaciteitsomschakeling uit te bouwen.
Zorg er voor dat Ic16 (74LS153) volledig vrij komt. Dit kan gebeuren door de nodige pennetjes uit het voetje te halen.
Zo moeten aldus pennetjes 2-1-9-10-11-14-15 van het Ic16 vrij gemaakt worden daar de pennetjes 2 en 14 voor iets anders gebruikt gaan worden.
Nu halen wij Ic12 uit het voetje en verplooien pennetje 10 zodanig dat dit later geen contact meer kan maken met het voetje, en plaatsen het Ic terug.
We moeten nu nog de volgende vebindingen maken:
Een draadje tussen pennetje 10 van Ic12 en Ic9 pen 1, en verder naar Ic16 pen 10, verder moet nu een draadje gelegd worden van Ic16 pen 9 naar het voetje van Ic12 pen 10.
Wij moeten ook nog een verbinding maken van Ic9 pen 3 naar Ic16 pen 11, en een verbinding van Ic4 pen 6 naar Ic16 pen 14 en van Ic4 pen 6 naar Ic14 pen 5.
Van Ic14 pen 6 word nu een verbinding gemaakt naar P12 pen 2.
Verdere inspiratie kan gezocht worden in het bijgeleverd schema "HIGH DENSITY DRIVE'S SUPPORT MODIFICATIE".

Hoe werkt het nu? Wanneer wij een 1.2 Mega byte drive willen aanspreken als 1.2 mega byte drive dan moet de clock op 16 Mhz worden ingesteld en de snelheid van de drive aangepast. Dit gebeurt door pen 6 van Ic4 laag te maken (bit 4 in het option byte). Verder moet de drive ook nog op dubbele density geschakeld worden, Dit gebeurd door bit 3 van het option byte laag te zetten.

Aldus moet het option byte er als volgt uitzien:0000 1000.
Indien deze drive gebruikt word als een standard 360 Kbyte drive dan ziet het option byte er als volgt uit:0011 0000.

Het drive tabel ziet er dan als volgt uit:

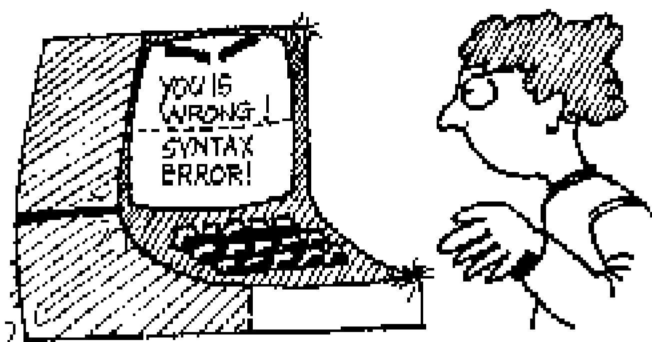
DRIVE	0,	1,	2,	3,	4,	5.	
#1FX0	5,	5,	5,	5,	5,	5.	Rom 5 is de 1793 driver
#1FX1	2,	2,	2,	2,	2,	2.	is de 2 de driver.
#1FX2	2,	2,	2,	2,	2,	2.	De drive is 0 .
#1FX3	0,	1,	0,	0,	0,	0.	Starting head.
#1FX4	1,	1,	2,	2,	2,	2.	Aantal zijden/schijf.
#1FX5	0A,	0A,	10,	12,	12,	1E.	Aantal sectors/track.
#1FX6	0,	0,	1,	1,	1,	1.	Eerste sectornummer.
#1FX7	38,	38,	30,	30,	10,	00.	Dualstep,ldens,sdens.
#1FX8	28,	28,	28,	28,	50,	50.	Cylinders/schijf.
#1FX9	4,	4,	4,	4,	4,	6.	Trackinterleave.
#1FXA	1,	1,	1,	1,	1,	2.	Headinterleave.
#1FXB	1,	1,	1,	2,	2,	2.	Sectorinterleave.
#1FXC	0,	0,	0,	0,	0,	0.	Stepping rate 6ms.
#1FXD	N.U.						
#1FXE	N.U.						
#1FXF	Checsum byte ingevuld door *DRTABEL [drive].						

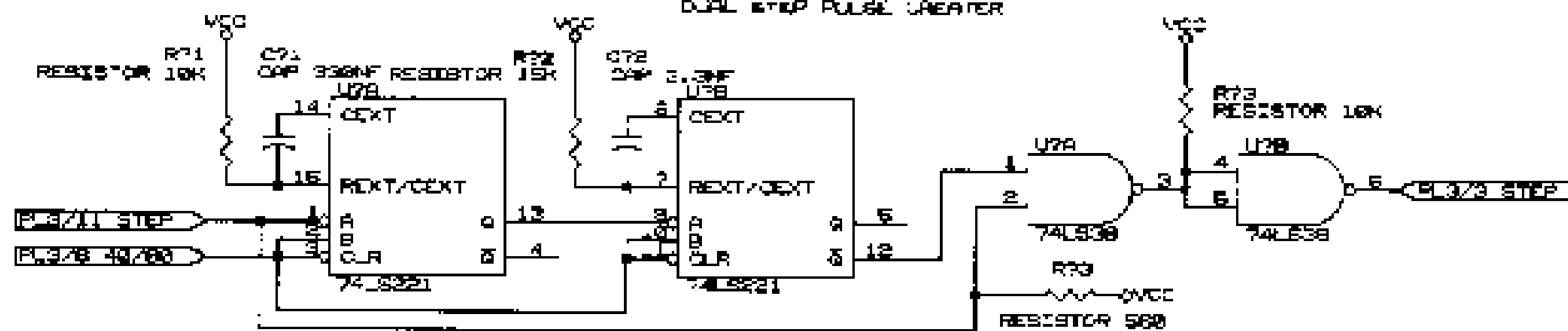
Hier zijn drive 0 en 1 respectievelijk de beide oude ATOM drives 0 en 2.
Drive 2 is een 320 Kbyte floppy systeem, met 16 sectors/track.
Drive 3 is een 360 Kbyte floppy systeem, met 18 sectors/track.
Drive 4 is een 720 Kbyte floppy systeem, met 18 sectors/track.
Drive 5 is een 1.2 Mega byte floppy systeem met 30 sectors/track.
Dit alles samengesteld uit 1 werkelijke drive.Het spreekt natuurlijk voor zich dat voor deze laatste 4 drive systemen alleen het multidirectory systeem in aanmerking kan komen.

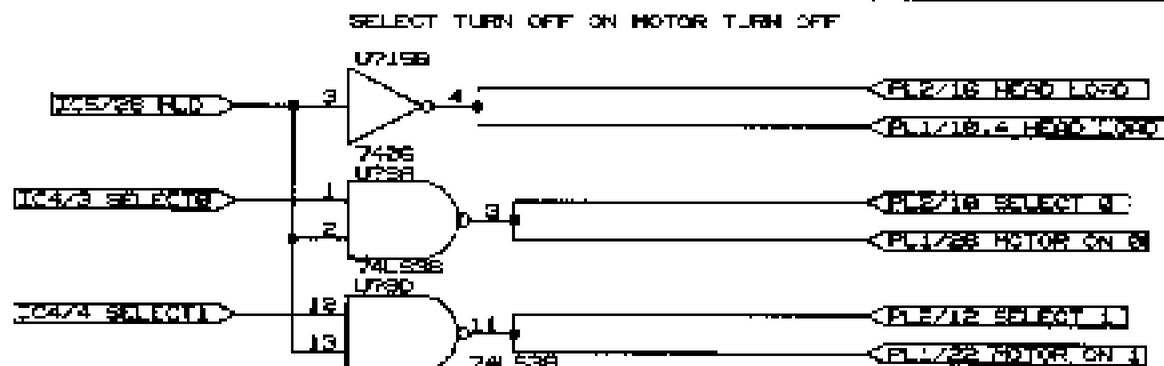
Indien de prijs voor de 3.5 inch floppys blijft dalen kan U zich binnen enkele tijd aan een update verwachten voor de 1.4 Mega byte en 720 Kbyte drives.Dezen worden nu reeds getest.

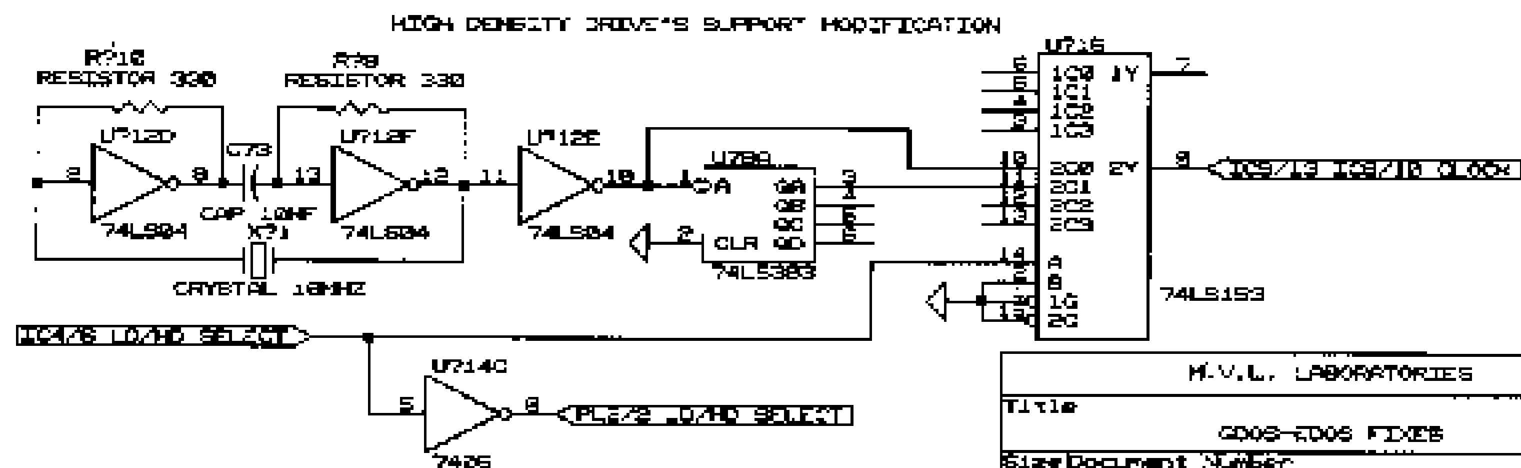
Tot zover mijne overpienzingen.
Beste groeten van :

Michel Van Leuven
Parsivalstraat 14
2580 St. Kat. Waver
Belgie
tel. 015-315082









M.V.L. LABORATORIES

Title		
GDOS-EDOS FIXES		
Size: A	Document Number: 1	REV
Date: February 5, 1989	Sheet 1 of 1	

COLOUR EMULATOR VOOR 80 KOLOMSKAART

Kleur op een monochroom scherm ... door Louis Eeger.

Het is mogelijk om met de 80 kolomskaart 8 verschillende kleuren in beeld te halen, echter alleen met een kleurenmonitor of kleuren TV.

Echter door phi 2 enigzins te veranderen (dit kan software matig) en er voor te zorgen dat de kleurenregisters van de EF9345 steeds juist staan, is het mogelijk om de rasterfrequentie van MONOCHROOM beeldbuis te beinvloeden. Dit heeft als gevolg dat de 'kleuren' razendsnel 'flikkeren' met een frequentie van 3.14*phi 2 Hz.

En wat gebeurt nu: door onze hersentraagheid krijgen wij het idee dat we kleuren zien !

De listing is vrij kort en tamelijk snel. Als je het programma runt wordt gevraagd waar de code moet komen te staan. Verder wordt ook gevraagd waar de 80 kolomssoft begint (VDU2.7 of VDU2.8). Voer de juiste adressen in.
Het programma schrijft de code weg en geeft daarna een demo.

Voor het geval dat je niet kunt wachten totdat je de regioschijf krijgt volgt hier het programma:

```
 10 REM Colour Emulator voor 80 kolomskaart
 20 REM (c) Atom Computerclub Nederland/Belgie
 30 INPUT "Waar komt de code te staan "P
 40 INPUT "Waar begint de 80 kolomsoft "T
 50 ?P=#20;P?1=T%256;P?2=T/256;P=P+3;@=0
 60 P!00=#0CF7D120;P!04=#65646956;P!08=#7250206F;P!12=#7365536F
 70 P!16=#20726F73;P!20=#74736E69;P!24=#656C6C65;P!28=#2E2E206E
 80 P!32=#48E1A52E;P!36=#A9F88320;P!40=#A2E18506;P!44=#F7D120A0
 90 P!48=#206A694A;P!52=#6F6C6567;P!56=#2074666F;P!60=#20686F6F
100 P!64=#656C6C61;P!68=#65682073;P!72=#EA20203F;P!76=#68DE00CA
110 P!80=#CA4CE185;P!84=#85BE22C2;P!88=#A0BE23A3;P!92=#208E20EA
120 PRINT "Aanroepen met  LINK #"&P-3'
130 PRINT "Druk toets voor demonstratie ... ";LINK #FFE3
140 REM Demonstratie :
150 LINK (P-3)
160 REM Bit 0,1 en 2 van #DE bevatten de kleur
170 FOR C=0 TO 7
180 ?#DE=C
190 PRINT "Colour Emulator voor 80 kolomskaart"'
200 NEXT C
210 END
```

```
DDDD      UU   UU     CCCC     KK   KK
DD DD     UU   UU   CC   CC    KK KK
DD   DD   UU   UU   CC         KKKK
DD   DD   UU   UU   CC         KKK
DD   DD   UU   UU   CC         KKKK
DD DD     UU   UU   CC   CC    KK KK
DDDD        UUUU      CCCC     KK   KK
```

Een nieuw spel voor de Acorn Atom door Roland Leurs.

Leidt Donald door het raamwerk en eet zoveel mogelijk vissen. Vermijdt de doodskoppen want hun lievelingsgerecht is canard a la nicoise (*). Let ook op de tijdbalk.
Je kunt Donald laten bewegen met de joystick of met de volgende toetsen :

```
                  [ - omhoog
z - links                       / - rechts
                  / - omlaag
```

Als je moe geworden bent kun je pauseren met de toets P of voor joystick-gebruikers de vuurknop. Het spel gaat verder als je een van bovengenoemde toetsen bedient.
Het spel gebruikt P-Charme, Gagsrom en Josbox (grmod) en het loopt van #2900 tot ongeveer #4300. Denk ook aan 2K voor de sprites.

Veel speelplezier met dit verkwakt leuke spel ....

met de vriendelijke groeten van Roland !

(*) Canard a la nicoise voor 7 personen:
een eend van 1 1/2 kg, 1 kg tomaten in vieren, 300 g ontpitte afgekookte olijven, 5 cl witte wijn, 5 cl cognac, 1 gestampt teentje knoflook, bouquet garni, 5 cl bouillon. Kooktijd: 55 a 60 minuten.
Braad de eend in boter goudgeel en voeg de tomaten, de olijven, en knoflook en de bouquet garni toe. Blus met de witte wijn, de bouillon en de cognac. Laat het sudderen tot het helemaal gaar is. Schep het vet af en serveer met de garnering eromheen. Smakelijk eten ! (Met dank aan Maurice op den Kamp - Born) .

```
*****************************************************************
*                                                               *
*    B B C   B A S I C   E N   C O M B I K A A R T              *
*                                                               *
*****************************************************************
```

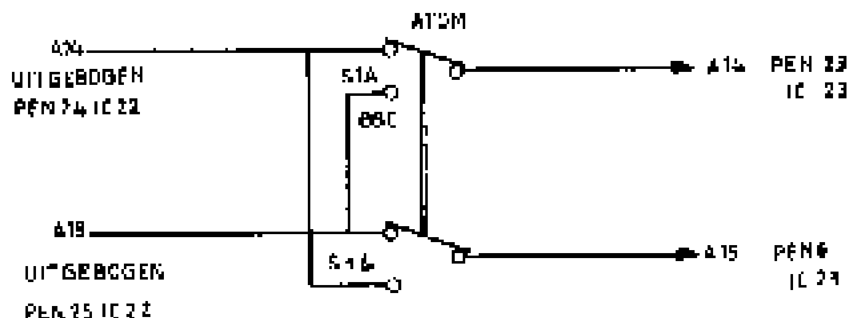
In dit artikel wil ik aangeven hoe men BBC Basic op een Atom met een combikaart kan implementeren. Het gehele bouwwerk is gebaseerd op een artikel van Ronald Boers in AN4-2.

Ronald heeft een eprom met de BBC-MOSROM in voetje 24 geplaatst en bij laadt de basic interpreter in RAM vanaf #4000 (Atom mode). Vervolgens drukt hij op de breaktoets en zet een schakelaartje om. En ziedaar ... BBC basic.

Er volgt in dat artikel een heel interessant verhaal hoe hij dat gedaan heeft en ik raad iedereen die BBC Basic in zijn Atom met combikaart gaat proppen aan om eerst dat artikel goed door te lezen. Ik geef alleen kort aan hoe je BBC basic aan de combikaart knoopt.

Omwisselen A14 - A15

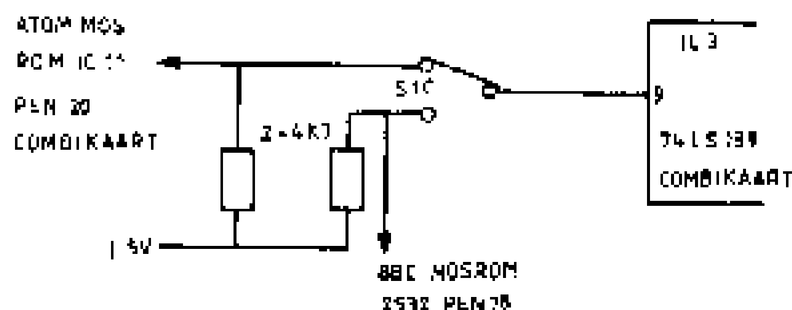
De BBC Basicinterpreter zit in BBC mode vanaf &8000. Als we A14 en A15 omwisselen verwisseld het gebied #4000 - #7FFF met #8000 - #BFFF (Atom mode). We moeten dus met een schakelaar deze gebieden verwisselen. Dit doen we met een driepolige omschakelaar waarvan 2/3 voor het verwisselen gebruiken, en wel als volgt:



figuur 1 : omwisselen A14 - A15

Omschakelen ATOM MOS - BBC MOS

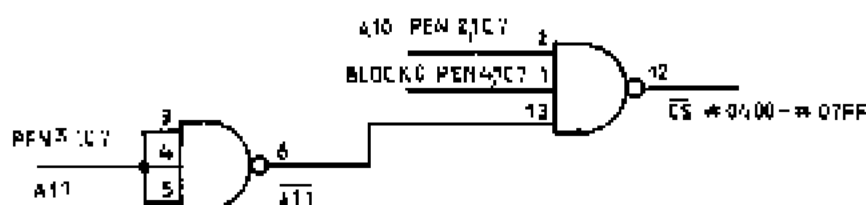
Het derde wisselcontact gebruiken we voor het omschakelen van de MOS ROMmen. De CS van ic 11 op de combikaart (CDEF rom) verbinden we via een weerstand van 4k7 met +5 V. Pin 9 van ic 3 op de combikaart buigt U uit het voetje. Deze gaat naar het middencontact van het derde wisselcontact van de schakelaar. De BBC MOSROM zit in een eprom (2532) in voetje 21 of 24. Pin 20 van deze eprom uitbuigen en via een weerstand van 4k7 naar pin 24. De cs-lijnen van beide MOS-ROMmen gaan naar de schakelaar, zie onderstaand schema:



figuur 2 : omschakelen ATOM - BBC MOSROM

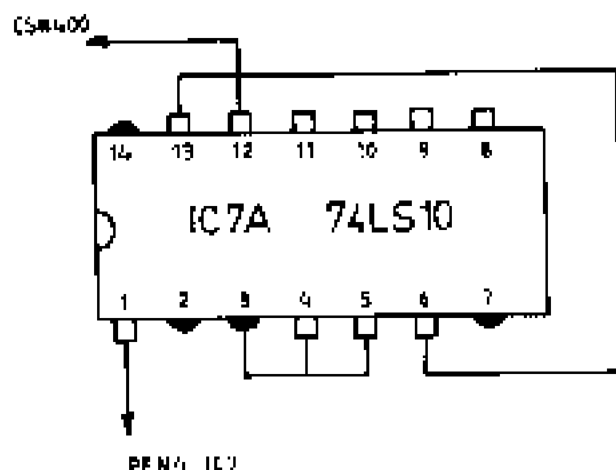
RAM op #400 #7FF

BBC basic gebruikt #400-#7FF als werkruimte. Als U RAM heeft van #0000 - #8000 kunt U dit deel overslaan. Er is immers al RAM van #400-#7FF. Anders moet U er een 74LS10 bijhalen. Plaats 2 2114's in de lege voetjes. Buig de CS pennen (pen 8) uit. Verbind de beide cs-lijnen met elkaar en met onderstaande decodeerschakeling.



figuur 3 : uitdecoderen #400 - #7FF

De gebruikte 74LS10 kan op IC7 gestapeld worden:



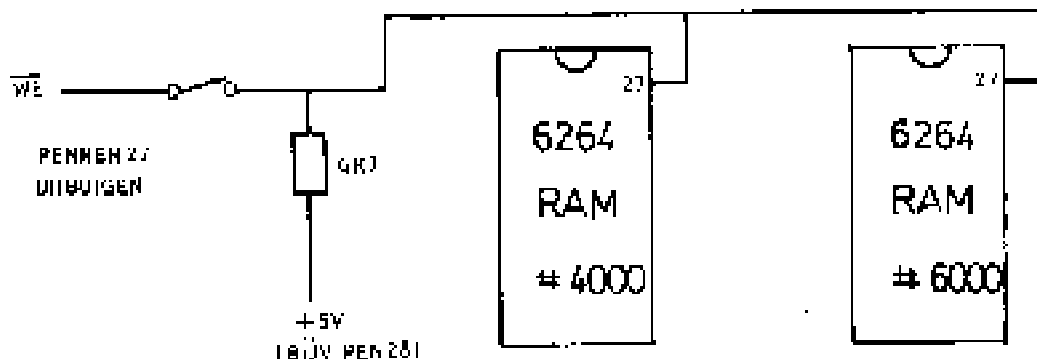
figuur 4 : 74LS10 op ic 7

Een andere oplossing is het gebruiken van een 6116, zie A.N. 7.1 blz 52.

WRITE PROTECT #4000-#7FFF

Uit ervaring heb ik geleerd dat write protect onmisbaar is omdat de BBC Basicinterpreter binnen de kortste keren kapot-geschreven is. Dit writeprotect realiseren we met een schakelaartje en een weerstand van 4k7.

Van de ic's 9 en 10 op de combikaart buigt U pin 27 uit. Bij een ic "plaatst" U tussen het voetje en de pin een schakelaartje. Verder komt er een weerstand van 4k7 tussen pen 27 en pen 28. Pen 27 van het tweede ram ic verbindt U met de andere pen 27. Zie ook onderstaande figuur:



figuur 5 : write protect #4000 #7FFF

Geheugen in BBC mode

Met een combikaart hebben we een lekker stuk geheugen voor BBC basic. Als er RAM zit van #0000 - #7FFF op de combikaart hebben we in BBC-mode geheugen van &800 t/m &3FFF (= 14 kB). Met 2*#1000 hebben we 12 kB tot onze beschikking.

Als nu iemand de 80 kolomsoft aanpast voor BBC-basic en als die 80 kolomsoft op &6000 (=#A000 Atom mode) komt te staan, dan krijgen we nog 8 kB erbij van het video geheugen. Dus totaal 18 resp. 16 Kb.

DOS

In AN4-2 schrijft Ronald Boers verder nog dat er geen gebruik kan worden gemaakt van de drive. Echter enkele maanden later maakte Marien van Westen de MINIDOS voor BBC basic bekend. Op het moment dat ik dit artikel inleverde bij de redactie had ik deze MINIDOS nog niet gevonden. Ik zal deze DOS proberen te vinden en als ik hem op tijd vind komt hij op de regioschijf. Mocht dit niet het geval zijn hoop ik dat iemand die deze MINIDOS wel heeft, dit even doorgeeft aan de redactie.

Gebruik BBC Basic

Laadt in Atom mode de BBC Basicinterpreter op #4000. Zet writeprotect #4000 aan en writeprotect #1000 uit (niet per se nodig). Druk dan de breaktoets in en schakel om. Laat break los en U kunt programmeren in BBC Basic.

Gebruikte onderdelen

1 schakelaar 3* wisselcontact
1 aan/uit schakelaar
3 weerstanden 4k7
1 2532 eprom met BBC MOSROM 3.0 (eprom programmeer dienst)
1 74LS10
2 2114 RAM ic's
draad

Artikel verwijzing

Meer informatie kunt U vinden in:

AN 4-2	blz 78	BBC Basic op de Atom
AN 4-3	blz 58	BBC kaartje
AN 4-5	blz 8	BBC MOS ROM V3.00
AN 4-5	blz 60	MINIDOS ROM voor BBC Basic
AN 7-1	blz 52	Geheugen van 0 - #7FF m.b.v. 6116

Veel plezier ermee ...

... met de vriendelijke groeten van Roland.

****
* Electron Nieuws *
****

van de redactie

Het bestuur van de regio Limburg heeft onlangs besloten om ook enkele bladzijden in dit blad te benutten voor de Electron. Deze computer heeft nog geen eigen club (in Limburg) en software kan alleen verkregen worden uit tijdschriften als Electron User (uit Engeland) en Asterisk (van de Big Ben Club).

Onder de Atomisten zijn er ook vele die een Electron bezitten en omdat we toch al een contactblad hebben kunnen we de Electron ook een kans geven. De bedoeling is niet dat we software van Acorn vrij gaan uitgeven, maar dat zelf gemaakte programmatuur wordt verspreid.

Verder is er natuurlijk ook hardware. Er zijn ontwerpen gemaakt voor diverse uitbreidingen die de Plus-1 omzeilen. Er is zelfs al een print gemaakt om een printer aan te sluiten!

Ook is er al een VIA print ontwikkeld met plaats voor 8*6522 VIA of een ander soortgelijk IC. Daardoor kunnen we uitgangspoorten realiseren die gelijk zijn aan die van de Atom. Dus om een eprom programmer te maken hoeven we alleen maar de software aan te passen voor de Electron, want de programmer hebben we al.

Een ander mooi idee is om de Atom Dos Controller print op de Electron aan te sluiten en de software voor een Electron Dos te maken waardoor we een drive aan de Electron hebben hangen.

Ideeen genoeg dus, als je interesse of ideeen hebt laat het dan even weten bij de redactie!

Namens de redactie, Roland Leurs.

Nog enkele tips :

In de regio Limburg behoren tot het drukwerkarchief drie volle klappers met informatie en programma's voor de Electron. Voorbeelden zijn :
een cad programma voor het ontwerpen van printen, spellen, informatie omtrent sideway Roms, handleidingen van programmeertalen en nog veel meer.

Als U nog geen diskdrive voor de Electron bezit, maar wel een Atom met BBC basic en diskdrive dan kunt U lange Electron programma's in BBC mode intypen en af en toe op disk stukken opslaan. Dit voorkomt woedeaanvallen als iemand de netspanning weghaalt.
Als het programma helemaal ingetikt is kunt U de listing oversturen naar de Electron met het programma AECOM. (zie regioschijf, opstarten met *AECOM).

Na een winter programmeren is het programma GOKKAST klaar.
Dit is een fruitmachine met twee beeldschermen.
Het programma GOKKAST bestaat uit 9 files. (ongeveer 40 Kbyte)
Dit programma kan men starten door de file GOKKAST in te laden en dit te runnen. De andere programma's worden dan automatisch ingeladen. Dit programma werkt ZONDER toolboxen.
Wel heeft het veel geheugen nodig, namelijk van #1000 tot #A000.

Nu volgt een korte beschrijving van de 9 files:

GOKKAST : Dit is het basic hoofdprogramma van #2900 tot #4F00.
MACH.T : Dit is machinetaal en de shapes
De machinetaal van het draaien van de rollen is van #5000 tot #5E00.
De machinetaal van het knipperen van de lampjes zit van #5E00 tot #6000.
De shapes zitten van #6000 tot #6500.
Van #6500 tot #6600 zit de machinetaal voor het verwisselen van de beeldschermen.
Van #6600 tot #6700 zit de machinetaal voor het geluid.
HO+NU : Dit is een basic programma voor HOLD en NUDGE van #9800 tot #9E00.
VDU : Is een programma in machinetaal voor in clear4 mode letters en cijfers te printen van #5000 tot #5800.
SCHERM : Dit is het hoofdbeeldscherm.
HO.LA : Dit is het beeldscherm hoger/lager van #6000 tot #8000.
PICTURE : Is het scherm SUPER DEAL.
CONTROL : Dit is het scherm waar de bediening van de toetsen op staat aangegeven.
PUNTEN : Is het scherm waar de waarden van de verschillende plaatjes staat aangegeven.

Nu volgt de werking van het spel:

Gokken. (De kolom rechts van het hoofdbeeldscherm)
--------------- ----------------------------------

Als het programma is ingeladen kan men door op de SPATIE balk te drukken de rollen laten draaien. Als twee de zelfde plaatjes vallen, op de middellijn worden deze automatisch vast gehouden. (NIET als deze in de eerste rol en de vierde rol zitten.) Als men de plaatjes niet wil vast houden dan moet men op toets "C" drukken. Men kan een plaatje wel vast houden (bvb de joker) door op de toetsen "1-2-3-4" te drukken.
Wanneer drie of vier de zelfde plaatjes naast elkaar vallen of een kers, dan krijgt men punten. Deze staan onder "WIN POINT". Men kan de punten pakken (TAKE WIN) of gaan gokken (GAMBLE). Deze twee lampjes gaan knipperen. Als men de punten pakt moet men op toets "T" drukken. Als men wil gokken moet men op de SPATIE balk drukken. Nu gaan de lampjes rechts een voor een aan en uit. Men moet nu een keuze maken hoeveel maal het aantal punten worden vermenigvuldigd.(4*-3*-2*-2*-1,5*)
Naarmate de vermenigvuldigings factor groter is, is de kans om

te winnen kleiner. Als het lampje van Uw keuze aan is moet men op de SPATIE balk drukken. Is nu goed gegokt dan worden de punten vermenigvuldigd onder "WIN PUNT". Is fout gegokt dan heeft men niks. Als men goed gegokt heeft en men wil door gokken dan moet men weer 2* op de SPATIE balk drukken, anders pakt men de punten door op toets "T" te drukken.

De kolom links van het hoofdbeeldscherm.

Als deze helemaal aan is kan men hoger/lager spelen. De lampjes gaan van beneden naar boven aan als er een overeenkomstig nummer valt op de middellijn, of als er een JOKER valt. Als er twee jokers vallen (ze hoeven niet naast elkaar te vallen, maar wel op de middellijn), dan gaat de de kolom helemaal aan en kan men hoger lager spelen.
De lampjes gaan uit als men goed gokt of als men punten pakt. Als men mis gokt de eerste keer, gaan de lampjes NIET uit.



Het beeldscherm hoger/lager.

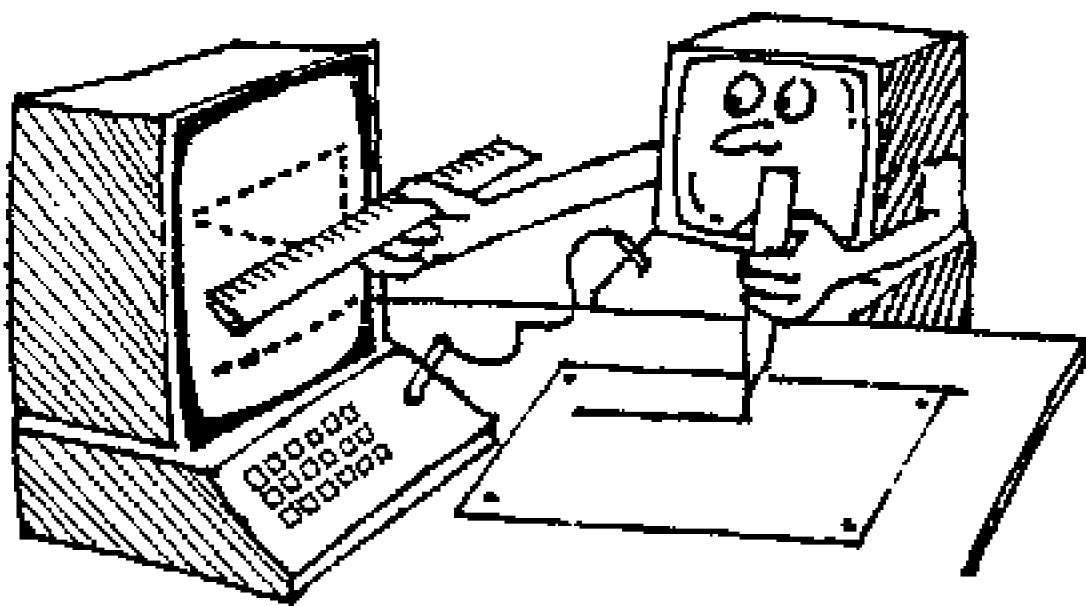
Er valt een kaart en als men denkt dat de volgende kaart hoger valt dan moet men hoger gokken (SPATIE balk) en als men denkt dat de volgende kaart lager valt dan moet men lager gokken (SPATIE balk).
Bij de rij onder gaat telkens als men goed gokt op volgorde een ander lampje aan. Dat is de winst. Deze winst kan men pakken door op toets "T" te drukken. (heel eventjes drukken) Als men punten pakt dan kan men terug bij het hoofdbeeldscherm gaan gokken (SPATIE balk 2*) of men kan de winst pakken (toets "T"). Als men HOLD of NUDGE pakt dan gaat de computer van het basic hoofdprogramma naar het basic programma HO-NU.

Hold.

Nu komt in het vak onder "WIN POINT" en "CREDIT" ,"8 hold" of "16 hold" te staan. Men kan nu de rollen 8 of 16 maal achter elkaar vast houden door middel van de toetsen "1-2-3-4". De rollen kan men los zetten door op toets "C" te drukken. Als men drie of vier dezelfde plaatjes naast elkaar heeft, kan men terug naar het hoofdprogramma door op toets "T" te drukken.
De winst kan men weer pakken of gokken.

Nudge.

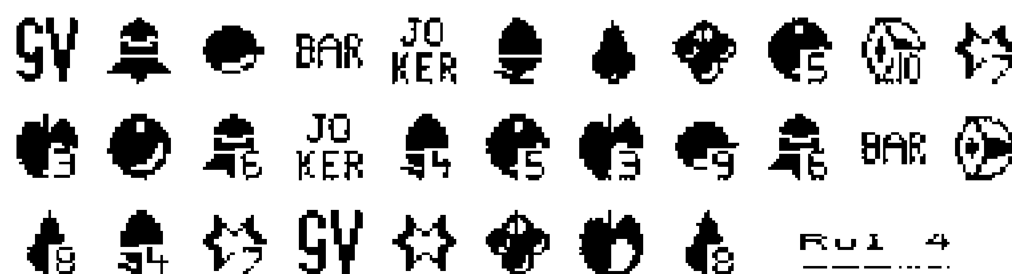
In het vak komt nu "8 nudge" of "16 nudge" te staan. Men moet nu de toetsen "1-2-3-4" gebruiken. De rollen kan men nu plaatje voor plaatje laten zakken. (van boven naar beneden), tot men drie of vier de zelfde plaatjes naast elkaar heeft. In fig.1 staan de plaatjes op volgorde hoe ze in de rollen zitten.
(In fig.1 van links naar echts is bij de rollen van beneden naar boven.) Als men geen plaatjes kan vinden die bij elkaar horen dan kan men door op de SPATIE balk te drukken de rollen een paar keer rond laten draaien tot men wel plaatjes vindt die bij elkaar horen.
Als men drie of vier de zelfde pllatjes naasyt elkaar heeft kan men terug naar het hoofdprogramma door op de toets "T" te drukken. Men kan nu weer de winst pakken of gokken.











Figuur 1.

TVWMmu80, een database of tekstbewerker.

door : Johan W.Hartog.

Het programma TVWmu80 is een combinatie van een eenvoudige tekstverwerker en een database
De editor is in feite een simpele database,gebaseerd op een editorprogramma gepubliceerd in num.6 (pag. 40) van de eerste jaargang van A.N.
Deze editor bezit in concreto records met een totale lengte van 64 karakters, zonder een veld-indeling.
De toevoeging "mu80" indiceert de aanpassing aan een Microline 80-printer.
Het programma heeft behoefte aan geheugen van #2900 to #8000, alsmede de aanwezigheid van de JOBBOX op voet 6 ,en werkt zo-mogelijk samen met BIG-BENNY om de tijd aan te geven "in de balk",indien althans het programma "knipcur" is geladen.
Indien dit het geval is ,dient overigens het woord "REM" in regel 20 verwijderd te worden.

Alle noodzakelijke instructies verschijnen op het scherm.
Het programmma zorgt ervoor dat tekst in kleine letters wordt uitgeprint.
* de insert-functie geeft de mogelijkheid ,in de textfile ,een complete regel toe te voegen,hetgeen vooral nuttig is bij gebruik als adressenbestand
* indien gebruik gemaakt wordt van het scheidingsteken (}), "shift-haak ",bestaat de mogelijkheid, een bestand met variabele velden te maken ; dan moet wel de printer-routine aangepast worden met name voor adres-labels.

Een 128-koloms-databestand kan in principe worden opgebouwd, zonder de gebruikersvriendelijkheid van het programma aan te tasten,de realisatie hiervan laat ik echter gaarne over aan de ATOM liefhebber.

J.W.Hartog.

noot van de redactie.
a. indien bij U de jobbox op een andere voet staat,wijzig dan de desbetreffende regels in het programma,waar deze wordt "voorgezet".
b. het programma heeft vrij veel geheugen nodig,laat dus niet zo heel veel ruimte over voor tekst;het is echter vooral van belang als leer- en ontwikkel-object.
BT.

DE BEGINNERSLOGE VAN ROLAND LEURS

```
BBBBB      AAAAA       SSSS      IIIIII     CCCC
BB  BB    AA   AA     SS  SS       II      CC  CC
BB  BB    AA   AA     SS           II      CC
BBBBB     AAAAAAA      SSSS        II      CC
BB  BB    AA   AA         SS       II      CC
BB  BB    AA   AA     SS  SS       II      CC  CC
BBBBB     AA   AA      SSSS      IIIIII     CCCC
```

deel 5

Nu we die feestdagen weer achter ons hebben liggen gaan we weer ijverig aan de slag met basic. Dit is deel 5 van de beginnersloge. Voordat ik begin met enkele huishoudelijke zaken wens ik U (misschien wel wat laat) een voorspoedig 1989.

Ik wil bij deze Maurice op den Kamp van harte bedanken voor het corrigeren van de teksten van de beginnersloge in 1988. Ik hoop dat ik ook dit jaar weer op zijn hulp kan rekenen.
Verder komen er weer misschien nieuwe beginners dit jaar. Het is natuurlijk onzin om weer helemaal opnieuw te beginnen met deze inleiding tot programmeren in basic. Hoe deze nieuwe leden en alle anderen aan de vorige afleveringen komen staat aan het eind van dit artikel.

In deze aflevering zal ik proberen om op een eenvoudige wijze het gebruik van arrays toe te lichten. Het lijkt allemaal heel moeilijk maar het valt toch wel mee. Belangrijk is dat U vertrouwd bent met gewone en floating point variabelen. Mocht er iets niet duidelijk zijn, bel of schrijf gerust. Mijn adres staat aan het eind van dit artikel.

D E A R R A Y

Inleiding:

Er doen zich in de praktijk veel situaties voor waarin we over een groot aantal variabelen willen beschikken. Stel bijvoorbeeld eens dat we met een voorraad administratie te maken hebben. Hierbij dienen we rekening te houden met artikelen en aantallen. Per artikel is er een aantal dat steeds varieert.
Zo zou de administratie van een Atom dealer met onderstaand programma kunnen worden bijgehouden:

```
10 REM Invoer gedeelte
20 INPUT "ATOM          = "A
30 INPUT "BBC BASIC     = "B
40 INPUT "COMBI KAART   = "C
50 INPUT "DISK PACK     = "D
   ...
   ...
```

```
1000 REM Bijwerk gedeelte
1010 INPUT "ATOM VERKOCHT      = "V
1020 A=A-V;PRINT "AANTAL ATOMS = "A
1030 INPUT "BBC BASIC VERKOCHT = "V
1040 B=B-V;PRINT "AANTAL BBC   = "B
     ...
     ...
2000 REM Voorraad op?
2010 IF A<10 THEN PRINT "ATOM BIJNA OP"
2020 IF B<10 THEN PRINT "BBC  BIJNA OP"
     ...
     ...
```

In dit voorbeeld ga ik er voor de eenvoud vanuit dat er nieuwe besteld moeten worden als er minder dan 10 stuks in voorraad zijn. We zien dat alle artikelen door een variabele worden weergegeven. De lezer zal geen enkele problemen hebben om te constateren dat dit programma volstrekt onbegonnen werk is. Want hoe moet dat met 100 artikelen. Ziet U de programmeur al typen: 1000 INPUT instructies, 1000 IF constructies, en dat alles met 2*26 variabelen...

Toch worden in veel bedrijven voorraad administraties gedaan met een computer (meestal een IBM of zoiets, maar als ik op regiosvonden rondkijk lijken veel Atoms ook op een cloon [of clown]). Hoe doen die dat dan???

Wel kijk:
- het is mogelijk om lange rijen gegevens in externe geheugens zoals cassette of floppydisk op te slaan en wel op zo'n manier dat er eenvoudige manier bewerkingen mogelijk zijn. Zo'n rij gegevens heet een 'file'.
- het is ook mogelijk om in het geheugen van de computer zo'n rij gegevens op te slaan. Deze gegevens zijn ook eenvoudig te bewerken. Deze rij gegevens heet een 'array'.
De files worden veel later nog behandeld en over de array ga ik het in deze aflevering hebben.

Ieder array element is in feite een doodgewone variabele waarmee we kunnen optellen, aftrekken etc. De notatie van een array element is in de vorm van AA(..) met op de puntjes een getal, variabele of uitdrukking. Het aantal te gebruiken array-elementen moeten we eerst reserveren met het statement 'DIM'. Achter DIM staat de naam van de array en het aantal elementen. Zie ook regel 10 in onderstaand programma.

Beschouw als eerste voorbeeld van een programma waarin een array voorkomt het nu volgende programma. Bij dit programma moeten 10 getallen worden ingevoerd. Van deze getallen moet dan de som worden bepaald en naderhand moeten alle getallen nogmaals afgedrukt worden.

```
10 DIM AA(10)
20 S=0:REM S staat voor 'som'
30 FOR K=1 TO 10
40   PRINT "Voer getal "K" in "
```

```
 50   INPUT G; REM G is een hulpvariabele
 60   AA(K)=G
 70   S=S+AA(K)
 80 NEXT K
 90 PRINT "De som is "S''
100 PRINT "De waarden waren :"''
110 FOR K=1 TO 10
120   PRINT K,AA(K)'
130 NEXT K
140 END
```

Een gedeelte van het resultaat ziet er zo uit:

```
 Voer getal  1 in ?14
 Voer getal  2 in ?5
 Voer getal  3 in ?10
   ...
   ...
De som is   736

De waarden waren :


     1      14
     2       5
     3      10
   ...
   ...
```

We starten met regel 10. Deze regel introduceert 11 variabelen waarvan we er trouwens maar 10 van gebruiken en wel met de uitdrukking

DIM AA(10)

De term DIM komt van het Engelse 'dimensions'. De 10 variabelen krijgen door deze uitdrukking de namen:

AA(0) AA(1) AA(2) AA(3) ... AA(10)

We zien dat deze variabelen als het ware een 'familienaam' hebben, namelijk de letters AA, en tussen de haakjes staat een nummer dat iedere variabele van een andere onderscheidt. Dit nummer noemt men ookwel de index een een array-variabele noemt men ookwel geindexeerde variabele. Zoals gezegd mag een array element hetzelfde gebruikt worden als een gewone variabele op twee uitzonderingen na:

NIET bij INPUT : INPUT AA(K) is fout en
NIET bij FOR/NEXT : FOR AA(0) = 1 TO 10 is fout, maar
FOR K=AA(0) TO AA(10) is toegestaan.

Dus het is wel goed om dit te gebruiken:

```
S=AA(1)+24
AA(2)=AA(2)*AA(3)
```

De index mogen we ook indirect door middel van een variabele, ja zelfs door middel van een uitdrukking aangeven. Zo zijn er dus drie situaties:

```
AA(3)    : index wordt direct door een getal aangegeven
AA(K)    : index wordt direct door een variabele aangegeven
AA(K*3)  : index wordt direct door een uitdrukking aangegeven
```

De index mag een floating-point getal/uitdrukking zijn maar dan telt als index het integer deel van dit getal.

Het programma toont ondermeer dat vooral de combinatie van arrays en FOR/NEXT lussen tot zeer krachtige concepten kan leiden. Laten we het eens zorgvuldig doornemen.

Regel 20 geeft aan de variabele S de waarde 0 en in regel 30 begint de FOR/NEXT lus. Steeds zal het programma 'Voer getal .. in?' op het scherm zetten. In regel 40 wordt de waarde ingelezen. Omdat dit niet rechtstreeks kan gebruiken we een hulpvariabele. In regel 60 krijgt het array element de waarde toegekend. In regel 70 wordt de som bijgewerkt.
Regel 100 zorgt voor de zin 'De waarden waarden :' en de lus die in regel 110 begint drukt alle getallen nogmaals af.

We hadden dus ook i.p.v. 10 variabelen heel gemakkelijk 1000 variabelen kunnen gebruiken zonder het programma enorm uit te breiden. We hoeven immers alleen maar regel 10 te veranderen in 'DIM AA(1000)' en de eindwaarden van de FOR/NEXT lussen moeten aanpassen.

FLOATING POINT ARRAYS

Ook voor de floating point hebben we arrays. Deze arrays moeten apart gedeclareerd worden. Dit doen we met het statement 'FDIM'. Met de floating point arrays kunnen we hetzelfde als met de gewone arrays. We kunnen ze zelfs direct inlezen met het FINPUT statement:

```
10 FDIM %AA(10)
20 FOR K=1 TO 10
30   PRINT "Voer getal "K" in "
40   FINPUT %AA(K)
50 NEXT K
 ...
 ...
```

MULTI DIMENSIONALE ARRAY'S

De hier besproken array's zijn allen een dimensionale array's. U moet dit zien als gewoon een rij getallen. We kunnen ook meer dimensies gebruiken. Er zijn hiervoor twee manieren. De eerste manier is vrij ingewikkeld en staat vermeld in het Atom boek van P. Smulders blz 77. Hier wordt met wat rekenwerk een trukje uitgevoerd.
De tweede methode is het gebruik van P-Charme, een uitbreidingsrom voor #A000. Een programma dat met P-Charme werkt begint altijd met het statement 'PROGRAM' gevolgd door een naam. Als we dit gedaan hebben kunnen we met bijv. drie dimensionale array's werken. Zo'n array kan als volgt uitzien: DIM RR(3,4,6).
Voor het gebruik van dit soort array's verwijs ik naar de

handleiding van P-Charme. Zelf zal ik later ook nog op deze uitbreidingarom terug komen.

DEMONSTRATIE PROGRAMMA'S

Op de regioschijf van dit blad staan enkele programma's die het gebruik van array's in de praktijk tonen. Het is voorlopig niet noodzakelijk dat U deze programma's helemaal begrijpt, maar probeer toch eens om de werking na te gaan.
De programma's zijn:

ZOEK1
ZOEK2
MIN&MAX
SORTEER

Ik geef hieronder een korte toelichting op de programma's:
ZOEK1 - zoekt een bepaald getal in een array
ZOEK2 - zoekt een bepaald getal in een gekoppelde lijst
MIN&MAX - geeft het kleinste en het grootste getal van een array
SORTEER - sorteerroutine met behulp van een array (bubblesort)

Tot zover mijn verhaal over arrays. Ik hoop dat het een beetje te volgen was. Het werken met array's is echt niet zo moeilijk als het lijkt. Gewoon zelf veel oefenen!

Succes ermee ...

... met de vriendelijke groeten van Roland.

Als U alle delen van deze cursus basic voor beginners zelf in huis wil hebben kunt U deze bij mij bestellen. De teksten zijn leverbaar als tekstfile voor de ED64/ED80 of afgedrukt op papier. De tekstfile kan op diskette of op cassette verstuurd worden.
Prijzen:

Diskette f 2,50 [diskette 1,- en 1,50 portokosten]
Cassette f 5,-- [cassette 3,- en 2,-- portokosten]
Afgedrukt f 5,-- [10 ct per copie en 2,-- portokosten]

Bestellen: Maak het bedrag over op gironummer 965809 t.n.v.
Roland Leurs
Nettenhoverkoestraat 5
6129 LH Urmond
met vermelding van 'Beginnersloge' en 'disk, tape of papier'. U ontvangt dan binnen enkele dagen de teksten en voorbeeldprogramma's.

EDIT80 voor TXP-1000 printer

Het artikel over de uitgebreide EDIT80 van G. v.d. Brandhof, bedoeld voor een Brother printer, heeft mij geïnspireerd tot het verbouwen van dit programma voor mijn eigen printer, een General Electric TXP-1000. Dat is een matrix printer en het printen van accenten e.d. gaat op andere wijze dan bij de Brother. Het werd al met al nog een forse operatie, voordat alle wensen waren gehonoreerd.
Om accenten te kunnen printen moet gebruik worden gemaakt van de Franse karakterset van de printer. Om ook met Duitse teksten uit de voeten te kunnen, heb ik tevens de mogelijkheden voor het printen van Umlauts ingebouwd.

Om dit alles mogelijk te maken zijn t.o.v. de standaard ED80 de volgende punt-commando's toegevoegd:

- .h Hulpspatie

Hiermee is het mogelijk om via '.hC' het karakter 'C' als spatie te gebruiken, met als doel dat bij het uitlijnen wordt voorkomen dat twee bij elkaar behorende woorden op verschillende regels terecht komen (ongewijzigd overgenomen).

- .b Bijzondere karakters

Na .b krijgen een aantal toetsen van het keyboard een andere betekenis.
Op het scherm wordt voor zover mogelijk met VDU80 reeds het gewijzigde teken (of iets wat er op lijkt) gepresenteerd, op de printer wordt het juiste karakter geprint.

- .f Franse karakterset

Na .f gaat de printer over op de Franse karakterset, waarmee het mogelijk is letters met accenten en ook een trema (") te printen.

- .g Germaanse karakterset

Na .g gaat de printer op de Duitse karakterset, voor letters met Umlaut e.d.

- .u USA karakterset

Door .u te geven wordt weer overgegaan op de standaard karakterset. Alle toetsen hebben dan weer de normale betekenis gekregen. Dit is tevens de default instelling na initialisatie.

Aan de .b, .f en .g commando's is verder de mogelijkheid gekoppeld voor onderstrepen en double width printen. Op het scherm wordt eveneens onderstreept, double width wordt weergegeven door invers video. Let op het uitlijnen bij double width, dit klopt niet meer.
Ook is de mogelijkheid voor superscript en subscript ingebouwd. Op het beeldscherm worden hiervoor de normale karakters gebruikt.
Na .u zijn bovenstaande mogelijkheden weer uitgeschakeld.
Bij het kiezen van .b dient een eventueel ingeschakelde Franse of Duitse karakterset eerst te worden uitgeschakeld met .u.

Bediening:
Om het geheel wat overzichtelijk te houden is de uitwerking van de verschillende mogelijkheden in tabelvorm weergegeven.

Toets		Ascii	.u		.b		.f		g	
			VDU	TXP	VDU	TXP	VDU	TXP	VDU	TXP
#	Sh-3	#23	#	#	underl	underl	underl	underl	underl	underl
$	Sh 4	#24	$	$	invers	d.w.	invers	d.w.	invers	d.w.
%	Sh-5	#25	%	%	norm	supers	norm	supers	norm	supers
&	Sh-6	#26	&	&	norm	subscr	norm	subscr	norm	subscr
@		#40	@	@	§	à	@	à	@	§
[		#5B	[	[	Ö	u	£	°	{	Ä
\		#5C	\	\	±	±	\	ç	\	Ö
]		#5D	]	]	$	¤	]	§	]	Ü
^		#5E	↑	^	‡	^	↑	^	↑	^
`	Sh @	#60		`	Þ	β		`		`
{	Sh-[	#7B	|	{	¶	φ	|	é	|	ä
|	Sh-\	#7C	|	|	•	"	|	ù	|	ö
}	Sh-]	#7D		}	◄	τ		è		ü
~	Sh ^	#7E	-	~	-	~	-	¨	-	ß

Zoals valt te zien is enig handwerk nodig bij het samenstellen van de tabel. Het vergt de nodige aandacht om de juiste codes in te typen, maar al met al valt er toch wel mee te werken.
Nog enkele bijzonderheden:

- De code voor een accent-circonflex en een trema worden voor de letter gegeven waarop het teken moet verschijnen (zoals bij een klassieke typemachine).
- Andere losse accenten kent deze printer niet, het is noodzakelijk hiervoor steeds de Franse karakterset te gebruiken.
- Het in- en uitschakelen van Underline, Double width, Superscript en Subscript is een toggle functie. De codetekens worden niet vervangen door spaties, omdat dit problemen geeft bij indices als in H_2O. Het uitvullen blijft correct functioneren.
- Superscript en Subscript sluiten elkaar uit; andere functies kunnen gelijktijdig actief zijn.

Verder heb ik nog een foutje opgelost in het .m commando. Dit commando bleek in ED80 niet te werken. De oorzaak was dat er naar een verkeerde plaats werd gesprongen. Door het toevoegen van een label (CC100) is dit opgelost.

Onder .b is in principe nog plaats voor twee bijzondere tekens, bij ^ en ~, maar de 4K is thans stampvol. De fraaie initialisatie tekst heb ik al opgeofferd, voor nog meer inkrimpen ontbrak mij de tijd.
De source heb ik van het nodige commentaar voorzien. Deze is daardoor wel erg lang geworden, tot ruim over #8000. Dit is voor VDU80 gebruikers echter geen probleem.
Tenslotte nog de Reference Card; deze is met veel geduld samengesteld door mijn zoon Jeroen. De kaart geeft een overzicht van alle mogelijke commando's voor de ED80 TXP.

Op het moment van verschijnen was de programmatuur nog niet aanwezig.
U kunt deze rechtstreeks aanvragen bij: A van der Veen
Merelweg 26
7442 CB Nijverdal
Tel.: 05486-14334
Op de regioschijven van ATOM NIEUWS nr. 2 zal bovengenoemde software worden bijgeplaatst.

ED80-txp Quick Reference Card

Bekijken tekst

S : (Start of text)	: Toon eerste pagina
Z : (End of text)	: Toon laatste pagina
N : (Next)	: Toon volgende pagina
P : (Previous)	: Toon vorige pagina
^ : (Roll up)	: Schuift tekst 1 regel omhoog

Cursorbesturing

[:	: Naar begin regel
] :	: Naar einde regel
H : (Home)	: Naar linkerbovenhoek
M : (Mark)	: Markeer cursorpositie
Escape	: Opheffen markering
Cursor toetsen	: Bewegen cursor

Wijzigen en corrigeren

E : (Edit)	: Toevoegen tekst
A : (After)	: Toevoegen bufferinhoud na cursor
B : (Before)	: Toevoegen bufferinhoud voor cursor
C : (Copy)	: Kopieer tekst in buffer
T : (Take)	: Verwijder en plaats in buffer
R : (Replace)	: Vervang gemarkeerd deel door bufferinhoud
I : (Insert)	: Voeg één karakter in voor cursor
X : (eXchange)	: Vervang één karakter
KCOPY	: Wis alles voor cursor
Delete	: Wist karakter op cursorpositie

Zoeken en Vervangen

/ : (Scheidingsteken)	: Teken dat niet wordt gebruikt in tekst
F/string1/	: Vind string tussen de scheidingstekens
F/string1//	: Zoek en wis string1
F/string1/string2/A	: Zoek en vervang zonder te vragen
F/string1/string2/	: Zoek en vervang string door string2
F:	: Type Y/N/ESC in (ESC is einde)

Disc acces

(: (Load)	: Laden tekstfile
) : (Save)	: Wegschrijven tekstfile
* : (Oscli)	: Executeer DOS/DOS commando (core on F1)

Output

O : (Output)	: Initieer output, wacht op O of P
O+O (of ander toe.)	: Output naar scherm
O+P (Output Printer)	: Output naar printer
SHIFT (Stop)	: Zet scherm stil en wacht op Ctrl
CTRL (Continue)	: Ga door met afdrukken op scherm
ESC (Escape)	: beeindig uitvoer

Diverse commando's

Q : (Quit)	: Beeindig programma en tel woorden
W : (Workspace)	: Geef op tot welk adres de tekst loopt
ED80 (Basic)	: Roep ED80 vanuit aan
TEXT (Basic)	: zet Basicprogramma op #0200
	: en in tekstfile op #7300

Bewerkingscommando's

.I	: Initialiseer puntcommando's
.lx (lines)	: Idem, geeft aantal regels/pagina op (default 63)
.rx (ruler)	: Regellengte
.px (page number)	: Start paginanummering
.p0	: Stop paginanummering (default)
.#	: Nummering gecentreerd
.wx (width)	: Breedte waarop paginanummer wordt geprint
.xA (exchange)	: Verandert bewerkingsteken van . naar A
.j (justify)	: Uitvullen aan (default)
.n (no justify)	: Uitvullen uit
.c (center)	: Centreren van 1 regel
.s (single)	: Enkele regelafstand
.d (double)	: Dubbele regelafstand
.ix (insert)	: Inspringen over x posities
.i (zonder getal)	: Inspringen uit
.a (zonder getal)	: Forceert nieuwe pagina
.gx	: Volgende x regels op zelfde pagina
.tx	: Spring x spaties in voor 1 regel
.m	: Bekijk een .LP voor regeltellen
.k	: Gebruikt losse vellen, meldt 'paper'
.ox	: Zendt getal x direct naar printer
.\	: Commentaar, wordt niet afgedrukt

Nieuwe commando's

.b (bijzonder)	: Schakelt bijzonder karakterset in
.g (german)	: Schakelt Duitse karakterset in
.f (french)	: Schakelt Franse karakterset in
.u (uit)	: Schakelt extra mogelijkheden uit
_	: Zet underline aan/uit
T	: Zet double width aan/uit
1	: Zet superscript aan/uit
4	: Zet subscript aan/uit

Speciale Karakterset's

Standaard	@	'	[	\	]	{	¦	}	^	-
Bijzonder	■	[illegible]	[illegible]	+	[illegible]	▌	`	|	^	▬
Frans	à	'	'	ç	§	[illegible]	ù	é	^	
Duits	§	'	Ä	Ö	Ü	ä	ö	ü	^	ß

Beschrijving RS-232 kaart en communicatieprogramma RSCOM.

De RS-232 kaart die hieronder beschreven wordt is niet de eerste, en waarschijnlijk ook nog niet de laatste die voor de ATOM gemaakt is.
Wat mijns inziens deze kaart van de andere onderscheidt, is ten eerste de volledig softwarematige besturing van de hardware en ten tweede de bijlevering van een redelijk pakket software om ook met de kaart te kunnen werken.
Dit houdt natuurlijk niet in dat voor iedere denkbare toepassing software bijgeleverd is, maar wel zijn er een eenvoudig terminalprogramma en enige kommando's als statement onder P-Charme.

Om te beginnen zal ik de hardware beschrijven.

Het hart van de RS232-kaart wordt gevormd door een 6551 ACIA
Voor de duidelijkheid: ACIA staat voor Asynchronous Communication Interface Adapter.
Dit is een IC dat het omzetten van paralelle in serieele data verzorgt, en andersom, want een ACIA is voorzien van een serieele ingang en een uitgang.
Hij verzorgt dus de omzetting van parallel in serieel met de toevoeging van start en stopbits en evt pariteit.
In de registers van de ACIA kunnen we alles wat met deze omzetting te maken heeft instellen en uitlezen, zoals baudsnelheid, karakterlengte en pariteit.
Ook kunnen we de toestand van de ACIA uitlezen in het statusregister.
De reden dat in dit ontwerp een ACIA wordt gebruikt, is dat op deze wijze het snelst en best gewerkt kan worden omdat de ACIA hiervoor speciaal ontworpen is.
Wanneer we met andere IC's werken, dan gaat het ook wel, maar de zaken waar niet in voorzien wordt moeten met extra software worden bijgewerkt, hetgeen ten koste van de snelheid gaat.
Met de ACIA en een geschikt programma is het zonder meer mogelijk om op een maximale snelheid van 19200 baud te werken.
Als ondersteuning van de ACIA is een 6522 VIA geplaatst, sommigen vinden dat misschien rigoreus, maar ik ben van mening dat voor de mogelijkheden waarin ik met het oog op de toekomst met dit ontwerp alvast wilde voorzien een VIA nodig is.
Deze (toekomstige) uitbreidingen betreffen de modembesturing door de ATOM, waarvoor de D poort van de VIA gereserveerd is.
De A-poort van de VIA wordt voor de volgende zaken gebruikt:

1: Besturing van de externe receiver-clock van de ACIA
2: Controle over het DCD-signaal (zie tekst)
3: Beldetektie (Ring indicator)

Om het externe clocksignaal op te wekken was een tweede kristaloscillator met een frekwentie van 1.8432 MHz nodig.
De kaart bevat dan ook twee kristallen, een voor de ACIA en een

voor de externe oscillator.
Met nog wat delers en een multiplexer is het mogelijk de externe klok in te stellen op 75 tm 9600 baud in 8 stappen. (steeds verdubbelen van de frekwentie)
De keuze voor een bepaalde snelheid wordt gemaakt door op de A-poort van de VIA in de bits 0, 1 en 2 een getal weg te schrijven. (0 tm 7) Het derde bit wordt gebruikt om de externe baudrategenerator aan of uit te zetten. (nul is aan, een is uit)
Al dit moois neemt op de bus 32 adressen in beslag, want ik heb de decodering zo gemaakt, dat er twee blokken van 16 bytes uitgekodeerd worden.
Dit is een volledig blok voor de VIA, en de ACIA (die maar vier adressen nodig heeft) wordt vier keer uitgekodeerd.
In mijn ATOM zit de eerste RS232 kaart op #B900, de VIA zit dus op #B900, de ACIA op #B910. Voor uitkodering van een tweede kaart zijn de adressen #B920 tm #B93F gereserveerd.

De software:

De software is in geassembleerde vorm op een paar bytes na een kilobyte lang (ik kreeg hem niet korter), en bevat de volgende kommando's:

1: FMT (formaat) Dient om de pariteit, de karakterlengte, en het aantal stopbits in te stellen. Alle kombinaties uit de datasheet zijn mogelijk.

syntax: FMT [1/2],[No/Even/Odd/Mark/Space],[8/7/6/5],[1/2]
Bijvoorbeeld om ACIA 1 in te stellen op NO PARITY, 8 databits en 1 stopbit is het kommando: FMT1,N,8,1

2: RST (reset) Geeft de ACIA een software-reset

syntax: RST [1/2] Voorbeeld: Reset ACIA 1 is: RST1

3: RX (ontvangsnelheid) Dient om een keuze te maken tussen interne of externe ontvangerklok en stelt de baudsnelheid voor ontvangst in.

syntax: RX [1/2],[I/E], voor interne en externe klok de baudsnelheid:
[intern: 50/75/110/134/150/3/6/12/18/24/36/48/72/96/192]
[extern: 75/150/3/6/12/24/48/96]

Vanaf 300 baud zijn de nullen te verwaarlozen, en tikken we enkel de cijfers in die voor onderscheid nodig zijn.
Voorbeeld; We willen ACIA 1 instellen op 1200 baud met gebruikmaking van de externe klok.
We tikken dan in: RX1,E,12

4: TX (zendsnelheid) Instellen van de zendsnelheid met als bijzonderheid dat wel de keuze voor interne klok gemaakt moet worden, hoewel externe zenderklok niet in te stellen is. (kiezen van externe zenderklok geeft een error) Dit heeft te maken met de algemene struktuur van de kommando's. Kommando's met meer dan een parameter verwachten als tweede karakter namelijk een letter, bij het FMT-kommando wordt met dit karakter de pariteit ingesteld. RX en TX bieden de keuze tussen Intern of Extern. (I of E)

syntax: TX [1/2],[I], Externe klok kan niet, na de I volgt de baudsnelheid, waarbij dezelfde keuzes mogelijk zijn als bij RX met interne klok.
Voorbeeld: Interne klok is verplicht, we willen ACIA 1 dan bijvoorbeeld instellen op 1200 baud
We toetsen dan in: TX1,I,12

5: TER (terminal mode) Laat de ATOM als terminal werken op de ACIA, keyboard-input wordt verzonden, ontvangen data wordt op het beeldscherm weergegeven.

syntax: TER [1/2] Voorbeeld: ACIA 1 als terminal inschakelen doen we met het kommando: TER1

Van tevoren moeten dan de zend en ontvangsnelheid ingesteld worden en moet met een FMT het juiste dataformaat opgegeven worden.

Het terminalprogramma test drie dingen achtereenvolgens:
1: Is er een toets ingedrukt? Zo ja, dan karakter versturen.
2: Is er data in de inputbuffer? Zo ja, dan uitprinten.
3: Is de CTRL-toets ingedrukt? Zo ja, dan wachten op een toetsaanslag.

Faciliteiten worden geaktiveerd door de CTRL toets in te drukken en vervolgens achter het vraagteken dat verschijnt uw keuze in te toetsen.
Ongeldige toetsen worden genegeerd, het ingetikte karakter verschijnt op het scherm, en de terminal bevestigt de toets met een piep, of in het geval u de P of de Q kiest verschijnt de prompt (>) achter het ingetikte karakter.
In terminalmode zijn de volgende faciliteiten in te stellen:
A: Printer aan of uit: CTRL,B of C
B: Keyboard echo op het beeldscherm: CTRL,E of F
C: Logmode aan of uit: CTRL,L of M
D: Verlaten terminal zonder reset van de ACIA
E: Verlaten terminal met reset van de ACIA
Optie B dient om te kunnen zien wat je zelf intoetst als de terminal waarmee je verbinding hebt geen echo's geeft. Dit is bijvoorbeeld het geval met twee ATOM's die communiceren, echter bij databanken zoals ATOMTEL is dit juist altijd wel het geval. het resultaat is dan dat alles wat je intikt dubbel op het beeld

komt. Na CTRL,F is alles weer normaal.
* Opmerking: Het kiezen van CTRL,D is hetzelfde als het inschakelen van TCPY (zie hieronder). Het nut van deze optie is dat nu de terminal tijdelijk verlaten kan worden en er gewoon in direct-mode gewerkt kan worden, bijvoorbeeld om een listing over te sturen.

6: TCPY (terminalcopy) Alle screenoutput gaat ook naar de ACIA, wel moet de ACIA dan eerst geaktiveerd worden met een TX-kommando.

syntax: TCPY [1/2] Voorbeeld: Kopieeren van screenoutput naar ACIA 1 wordt gedaan na het kommando: TCPY1

7: NCPY (nocopy) Beeindigt TCPY.

syntax: NCPY [1/2] Voorbeeld: Om te stoppen met kopieeren naar ACIA 1 is het kommando: NCPY1

Tot zover de kommando's

Verdere beschrijving van de software:

De software is gemaakt voor twee RS232 kaarten, die volgens de dokumentatie hierboven zijn gebouwd.
Wanneer een tweede kaart bijgestoken wordt, hoeft de software dus niet te worden aangepast.
Dit is bereikt door zowel de ACIA als de VIA geindexeerd te adresseren (ACIA,X en VIA,X)
Dit gaat als volgt: De eerste parameter die bij alle kommando's dient te worden opgegeven is het poortnr, dit kan zijn 1 of 2.
Hieruit wordt de offset berekend, voor poortnr 1 is de offset #00, voor poortnr 2 is dit #20 (32 decimaal).
De software verwacht dus dat als de tweede ACIA geplaatst wordt, dat deze #20 hoger in het geheugen zit, dan de eerste.
De geheugenlokatie met de naam OFFSET dient om dit getal op te slaan.
Bij elke lees of schrijfopdracht naar de kaart wordt het X-reg geladen met de waarde van OFFSET, zodoende "weet" het programma steeds met welke poort er gewerkt moet worden.

Volgens mij moest het mogelijk zijn om het grootste deel van het programma te gebruiken voor andere kaarten, waarop alleen maar een 6551 ACIA zit. Dit dan uiteraard zonder de split-speed faciliteit (verschillende zend en ontvangstsnelheid), omdat de ACIA dat zonder externe baudrategenerator niet aankan. Zolang poortnr 1 gebruikt wordt is de OFFSET nul, dus wordt er "absolute" geadresseerd. Wanneer u in de source het adres van uw ACIA aangeeft, en altijd een 1 achter de kommando's zet zal het programma dit verschil niet merken. Nu zou ik niet durven beweren dat de soft universeel is, zonder dit eerst uit te testen, en dat heb ik dus ook gedaan, met onderstaand resultaat.

De kommando's RX en TX werken, zolang er maar voor de interne klok gekozen wordt. Het kiezen van de externe klok heeft geen ERROR tot gevolg, het programma kijkt namelijk niet of de VIA werkelijk aanwezig is maar hij schrijft gewoon op het adres waar volgens de source de VIA zit de juiste informatie weg.
Als op dat adres toevallig een ander IC zit dan gaat er denk ik iets fout, dus kun je dit het beste naar een leeg stuk in het D blok laten wijzen.
In de ACIA wordt alleen bit 4 van het controlregister op nul gezet, als voor de externe baudrategenerator gekozen wordt.
Als er dus geen externe klok aanhangt, zal de ontvanger hierna niet meer werken. Wanneer er wel een externe klok aanwezig is, zal de ontvanger werken op de baudrate van die externe klok, en niet op de opgegeven snelheid, maar dat lijkt me nogal logisch.
De kommando's RST en FMT werken zonder meer met een 1 als poortnr, want ze schrijven enkel naar de ACIA.
Het kommando TER werkt ook, maar hierbij past een waarschuwing.
Lees het onderstaande eerst aandachtig door en kijk of er in je ATOM op de aangegeven adressen VRIJE geheugenruimte is.
Zoals gezegd werkt de ontvanger onder interrupt.
De IRQ-routine doet het volgende: in #C0 tm #C8 staan vier pointers, die wijzen naar de pagina's #0600 en #0700.
Twee ervan zijn de leespointers, de andere de schrijfpointers.
Per ACIA is een lees en een schrijfpointer nodig, maar omdat de software voor twee ACIA's gemaakt is zijn er vier pointers.
Voor ACIA 1 is dit #C0 en #C2, voor ACIA 2 #C4 en #C6.
Wanneer er een byte van de ACIA komt, wordt de schrijfpointer met 1 verhoogd en wordt het ontvangen byte weggeschreven op het adres waarnaar de schrijfpointer dan wijst.
Voor ACIA 1 staat de schrijfpointer op page #0600, voor ACIA 2 is page #0700 gereserveerd.
Mensen die hier iets anders hebben staan zullen deze buffers ergens anders moeten neerplanten.
Dit is te wijzigen in de subroutine BUFFERS.
Verder staat het testgedeelte van de IRQ-routine op #0E00, hier wordt getest of het interrupt van ACIA 1 of 2 afkomstig was, waarna naar de afhandelingsroutine gesprongen wordt.
Ik heb voor deze opbouw gekozen omdat ik ooit nog eens een universele IRQ-afhandelingsroutine wil maken, die alle IC's die een interrupt kunnen genereren afloopt, kijkt welke het interrupt veroorzaakt heeft, en dan naar een subroutine springt die het betreffende interrupt afhandelt.
In de initialisatie-routine wordt de IRQ-vector (#204+#205) op #0E00 gezet, ook hier geldt: dit is te wijzigen. Ook moet dan de opdracht [illegible] #0E00 in de laatste regels van het programma verzet worden naar het nieuwe adres.
Verder is de logpointer (#C8+#C9) als beginwaarde op #5000 gezet. dat wil zeggen dat wanneer de logfaciliteit ingeschakeld wordt, de ontvangen data vanaf #5000 in het geheugen wordt gekopieerd.
Jammer als er dan op #6000 P-Charme tabellen en dergelijke staan, want 4k (van #5000 tot #5FFF) is snel vol. Wijzigen van het

startadres voor de logfunktie is de remedie. In de subroutine AOFF wordt dit adres geinitialiseerd.
Bij het uittesten van de soft op een andere ATOM kwam ik nog een ander puntje tegen, wat voor problemen zou kunnen zorgen.
Na het assembleren van het programma voerden wij een RX kommando uit, en meteen begon het hele scherm te knipperen, na een reset van de 80-kolomskaart was alles weer in orde. Het probleem zat volgens mij in de zero-page, en dat bleek ook te kloppen, want nadat de tabel in zero-page (waarin de lees en schrijfpointers zitten) naar #80 was verzet was het probleem over. Het label :BASE in de listing geeft aan waar de tabel in zero-page moet beginnen. Dit label stond bij mij op #C0. De versie van de 80-koloms-soft, waarbij dit optrad was 3.1, zelf heb ik op het moment versie 3.7, en daarmee heb ik het probleem niet gehad.

Er is nog een puntje in het programma dat extra toelichting verdient. Bij het testen bleek dat de interrupts niet altijd goed werden afgehandeld. In terminalmode was dit probleem er niet, maar na het kommando TCPY1 trad steeds vrij snel een ERROR op, waarbij een merkwaardigheid was, dat hoe hoger de baudrate was, hoe minder fouten er kwamen. Op 1200 bd full duplex was de situatie echter zodanig dat hier iets aan gedaan moest worden.
Na redelijk wat zoekwerk kwam ik tot de konklusie, dat de fout in de schrijfroutine voor de ACIA moest zitten. In de oorspronkelijke vorm was deze voorzien van een testloop, die alleen wachtte tot het TX-register leeg was, en daarna het byte verstuurde. Deze loop las steeds het statusregister uit en keek of bit 5 hoog was (TX-reg empty). Op relatief lage snelheden kan dit vrij lang duren, wat betekent dat het statusregister enige duizenden keren gelezen wordt. Volgens de datasheet van de 6551 heeft lezen van het statusregister tot gevolg, dat de interrupt-flag op nul gezet wordt. Mijn oorspronkelijke interruptroutine keek eerst naar bit 7, om te zien of er een interrupt stond.
Wanneer het programma stond te wachten, tot het TX-register leeg was, en er kwam op dat moment een interrupt binnen, dan werd de instruktie, waarmee de processor bezig was eerst afgemaakt. Het zou in zo'n geval kunnen dat die instruktie de leesopdracht van het statusregister is. Dit heeft tot gevolg dat het interrupt gewist wordt, de 6502 springt na het afmaken van de instruktie naar de IRQ-vector en daag!! De eerste oplossing die ik bedacht was om de interrupt-routine niet meer eerst te laten testen of de interrupt-flag van de ACIA er nog staat, maar ook te testen op bit 3. (RX register full) Dit gaf in eerste instantie een grote verbetering, die mij de overtuiging gaf dat ik in de goede richting zat te zoeken. Toch was het nog niet perfekt, en ik denk dat ik zonder een logic analyzer nooit in staat zal zijn om precies uit te vissen waar het knelpunt zit. Nu heb ik wel de beschikking over een scoop, en die heb ik toen eens aan de IRQ gehangen. Het bleek al snel dat de breedte van de IRQ-pulsen niet erg konstant was, en soms leken ze haast naaldvormig. Ik heb op grond van die informatie de volgende theorie ontwikkeld:
Wanneer de IRQ-puls, die veroorzaakt wordt doordat er een byte

ontvangen is (RX-reg full) nagenoeg samenvalt met het lezen van het statusregister, wordt deze IRQ meteen weer gewist, de puls wordt dan naaldvormig. Aangezien de CPU enige tijd nodig heeft om een interrupt te detekteren vormt dit een probleem. De ATOM zal geen IRQ detekteren, maar zou omdat deze puls eigenlijk als stoorpuls optreedt, toch iets kunnen doen wat hij niet geacht wordt te doen. Wel is het gevolg dat er een byte verloren gaat (het RX reg wordt niet gelezen), en mijn vermoeden is dat deze naaldvormige IRQ-pulsen de problemen veroorzaakt hebben.
De remedie was zien te voorkomen dat een interrupt samenvalt met een leesopdracht van het statusregister, bijvoorbeeld door het aantal leesopdrachten van het statusregister te beperken.
Ik heb dit eenvoudig kunnen realiseren door timer 1 van de VIA te gebruiken. Door deze timer in one-shot mode te gebruiken om een timeout op te wekken kunnen we de ATOM eerst op deze timeout laten wachten, voor hij het statusregister van de ACIA uitleest. Mits de timeout juist gekozen wordt heeft dit geen gevolgen voor de tranmissiesnelheid. Een korte berekening leert dat bij de maximale snelheid van 19200 bd er 1920 bytes per seconde binnenkomen. (8 databits en 1 start en stopbit) Dit is dus 1 byte per 500 uSEC. Ik heb de timeout op de helft van deze tijd ingesteld, en na deze "truuk" werkte alles foutloos. Op mijn scoop hadden de IRQ-pulsen nu een konstante breedte van 50 uSEC.
De schrijfroutine voor de ACIA leest nu dus het statusregister met tussenpozen van ongeveer 256 microseconden uit. Dit betekent voor bijv 1200 bd dat het statusregister maximaal ongeveer 32 keer wordt uitgelezen, hetgeen een grote verbetering is. Verder biedt dit aan eenieder de mogelijkheid om met deze tijden te stoeien. Het is namelijk mogelijk om deze tijd in te stellen.
Bij het geven van een RX, TX of TER kommando wordt de tijd op 256 uSEC ingesteld, door na dit kommando register 7 van de VIA (Timer 1 High latch) te wijzigen kan deze tijd tussen nul en 65 mSEC ingesteld worden in stapjes van 256 uSEC. Het programma leest namelijk de tijd uit de latch, en schrijft die in de counter (vijfde byte van de VIA), waardoor de timer gestart wordt. nadat de timer op nul gekomen is gaat de routine dan verder met het lezen van het statusregister van de ACIA. Is de transmitter reg niet gereed, dan wordt het hele gebeuren opnieuw opgestart, net zolang tot bit 4 in de ACIA hoog is. Een berekening over het snelheidsverlies is als volgt: De maximale vertraging is, afgezien van de looptijden van het programma gelijk aan de ingestelde vertraging van T1. Als deze standaard 256 uSEC is en bij elke byte treedt de maximale vertraging op, dan duurt het transport van een file van 4k 4096x256 uSEC, ofwel ongeveer 1 seconde langer. Volgens mij is het snelheidsverlies dan ook niet te merken, hoewel dit procentueel oploopt, wanneer er hogere snelheden toegepast gaan worden.
Experimenteren met deze vertraging gaat als volgt:
Na een FMT(1,N,8,1) en een [illegible](1,1,12) opdracht en het kommando ECHO1. Alle beeldschermoutput komt nu met 1200 bd op het scherm. Tik nu bijvoorbeeld V?7=#60, V is hier het startadres van de VIA, en kijk wat er gebeurt! Een mogelijke toepassing van dit grapje kan

zijn het vertragen van te verzenden informatie, als de ontvanger een lage verwerkingssnelheid heeft.
Een VIA is wel nodig voor deze routine, maar in de listing wordt deze aangegeven met VIA2, wanneer u geen ACIA-kaart heeft volgens bijgaand schema, kunt u hiervoor gewoon de system-VIA op #B800 gebruiken. Als VIA2 niet wordt aangewezen werkt het programma niet!

Uitbreiding van de software:

De software is nog uit te breiden met vele faciliteiten, zoveel dat ik ze hier niet kan opsommen. Wel biedt deze software een paar aanknopingspunten om op verder te borduren.
Zo is het uitlezen van de ontvangen data een voorbeeld.
Bewust heb ik voor de bufferstructuur gekozen als zijnde een compromis tussen snelheid en efficiency.
Het is natuurlijk het meest efficient om je data meteen op de goede plaats te zetten, met een eigen interrupt-routine.
Aan de andere kant is de bufferstructuur ook vrij snel (hoewel ik het programma nog niet specifiek op snelheid geschreven heb), en het is een universele oplossing, die door en door getest is, en die nu probleemloos blijkt te werken.
En als ik later nog eens iets anders wil doen met de binnenkomende data, dan hoef ik niet meer in de interruptroutine te vogelen, maar alleen het uitleesprogramma te wijzigen.
Het uitlezen van de buffer gaat met de leespointer, wanneer het LSB van de leespointer niet gelijk is aan het LSB van de schrijfpointer, dan is er data in de buffer aanwezig.
Uitlezen gebeurt door de leespointer met een te verhogen, het karakter op dit adres te lezen, en te kijken of het LSB van de leespointer gelijk is aan het LSB van de schrijfpointer.
Is dit nu het geval, dan is de buffer leeg, anders moet er verder gegaan worden met uitlezen.
Met de bytes uit de buffer kan een ieder vervolgens zijn gang gaan.
Een voorbeeld van een machinetaalprogramma om de buffer uit te lezen en de gelezen data op het beeldscherm te printen vindt u in het terminalprogramma. (vanaf label :NOKEY tm label :NIKS)

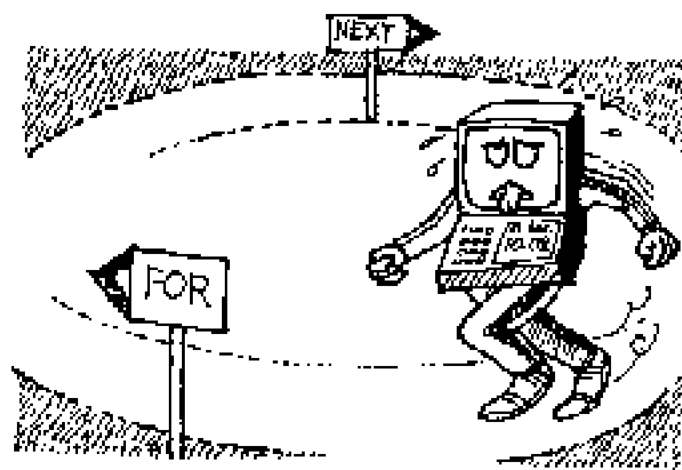
Het schrijven naar de ACIA gaat niet onder interrupt (zou wel kunnen via bit 4 van het statusregister), maar hiervoor heb ik de subroutine WRACIA gemaakt. Deze routine is hierboven al uitgebreid beschreven. Rest nog te vertellen dat de waardes van de A, X en Y-registers gelijk blijven aan de waardes die ze hadden bij binnenkomst in de routine en dat hij werkt met geindexeerde adressering, en dus geschikt is voor gebruik met meerdere ACIA's. Deze routine kan bijvoorbeeld ook worden gebruikt om een seriele printer aan te sturen.

Iets wat in de software (nog) niet is verwerkt is de kontrole over het DCD-signaal (Data Carrier Detect).
Dit signaal wordt bij kommunikatie met een modem gebruikt om aan

te geven dat er een carrier aanwezig is van het modem wat opgeroepen wordt, met andere woorden de verbinding staat er.
De ontvanger van de ACIA heeft dit signaal nodig om te kunnen werken (zie de datasheet).
DCD is een laag aktief signaal dus nul is true, een is false.
Als we nu een modem aan de ACIA zouden hangen, dat de DCD niet constant aktief houdt, dan kunnen we alleen maar werken als de verbinding er staat.
Kommando's aan het modem geven kan dan niet als er geen DCD is.
Natuurlijk is dit met de soldeerbout te verhelpen, maar dat vond ik voor dit projekt wat te primitief.
Op de kaart heb ik dus een voorziening gemaakt om DCD toch aktief te kunnen maken, onafhankelijk van de toestand van de interface.
Op de VIA heb ik hiervoor bit 4 van de A-poort gebruikt.
Dit bit staat als uitgang geprogrammeerd, en zit samen met het DCD-signaal op een AND-poort aangesloten.
Als bit 4 van de VIA laag gemaakt wordt, zal DCD ook laag worden, dus aktief, maken we bit 4 hoog, dan hangt het van de "echte" DCD af of DCD op de ACIA aktief wordt of niet. Bij initialisatie van de ACIA wordt bit 4 van de VIA als uitgang geprogrammeerd, maar verder wordt er niets meer mee gedaan.
Ook de ring-indicator op bit 7 is niet geprogrammeerd, maar voor geinteresseerden is de mogelijkheid aanwezig. De reden dat het ring-indicator signaal via een optocoupler wordt aangesloten is gewoon dat ik in in alle gevallen wilde voorkomen dat er plus of min 12 volt op de VIA terecht zou kunnen komen. Het ring-indicator signaal is ook laag aktief, dus nul is: er wordt gebeld, een is: er wordt niet gebeld.

De volgende uitbreiding, waaraan al gewerkt wordt is een viditelprogramma voor de 80 kolomskaart. Ook zit ik ik nog te peinzen over een eenvoudige autotransmit. Daarover misschien later meer. Voor vragen en / of opmerkingen ben ik op onderstaand telefoonnummer te bereiken.

Leo Gijssel
Ruisdaelstr 6
4461 AD Goes
Tel: 01100-32557
Regio Zeeland



Het schema op orginele grootte (A3) is te bestellen bij H. Sanders.
Zie printservice.

AUTOTRANSMIT OP RSCOM

Hierbij nog even een nagekomen stukje over RSCOM.
In het vorige artikel stond, dat aan een eenvoudige autotransmit gewerkt werd. Dat werk is ondertussen afgerond, en de versie die nu in de club verspreid gaat worden, (versie 2) heeft de autotransmit-funktie er al bij.
Er is hier duidelijk sprake van een eenvoudige autotransmit.
De gebruikte editor, om teksten voor verzending gereed te maken is de ingebouwde terminal van RSCOM. Waarschijnlijk zal het met edit-80 e.d. ook wel gaan, maar die heb ik al sinds tijden uit mijn ATOM verwijderd, omdat Wordstar onder CPM mij stukken beter bevalt, en omdat mijn schakelkaart vol dreigde te raken.
De beperkingen die worden opgelegd doordat de terminal als editor wordt gebruikt zijn mijns inziens niet erg groot en een voordeel van deze methode is, dat er maar een programma nodig is.
De belangrijkste beperking is dat gemaakte tikfouten "niet echt" hersteld worden. Daar bedoel ik mee dat alles wat ingetikt wordt meteen in het geheugen van de ATOM gekopieerd wordt, wanneer u dus tikfouten maakt, en ze vervolgens herstelt met de DELETE toets, dan staat de fout in het geheugen, gevolgd door een DELETE-teken (#7F), gevolgd door het goede karakter. Wanneer de opgeroepen databank de code #7F niet herkent als zijnde een DELETE, dan staat je bericht er een beetje lullig in. ATOMTEL herkent de DELETE's wel, en zo hoort dat ook natuurlijk, maar hoe dat bij andere databanken (bijv FIDO) is weet ik echt niet.
Deze editmogelijkheid is dan ook niet bedoeld voor zeer intensief gebruik, maar voor korte berichtjes is hij goed te gebruiken.

Hoe werkt het:
De speciale funkties van de terminal, zoals printer aan/uit zijn uitgebreid met een autotransmit, die wordt aangeroepen met een A. Na het intikken van CTRL,A wordt de inhoud van het geheugen vanaf de start van het LOG-geheugen verzonden, totdat in de tekst een #FF staat. Om alles compatibel te houden is het noodzakelijk dat voor dit teken een #0D (CR) komt. Het programma verzorgt dit normaal gesproken zelf, maar als u de files op een andere wijze wilt aanmaken zult u hiervan op de hoogte moeten zijn.
Tot zover het verzenden, maar voor we iets kunnen verzenden zullen we eerst moeten zorgen dat er ook iets staat om te verzenden.

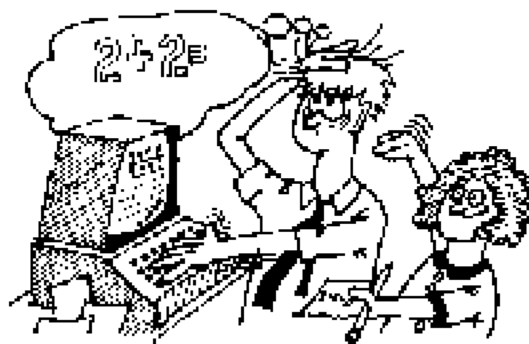
Het aanmaken van een file mbv de terminal.

Bij het nalezen van mijn beschrijving van de terminal van RSCOM bent u ongetwijfeld de LOG-funktie tegengekomen. Deze wordt normaal gebruikt om ontvangen data te kopieeren in het geheugen, vanaf een tevoren in het programma vastgesteld beginadres.
In de op de clubschijf geleverde versie staat dit adres op #5000 (zie tekst bij RSCOM). In de oorspronkelijke versie, zonder autotransmit, werd dit adres vastgesteld in de subroutine #AFF.

Nu is hiervoor een aparte subroutine gekomen die alles regelt voor wat betreft de LOG funktie, INITLOG genaamd. Wij gaan deze LOG-funktie gebruiken om ons eigen tikwerk in het geheugen van onze ATOM op te slaan. Wanneer uw modem namelijk in Local mode staat, dat wil zeggen er is geen verbinding met een andere computer, dan moet het mogelijk zijn om de uitgezonden data zelf terug te ontvangen. Dit wordt in datacommunicatie-kringen een Local Loopback genoemd. *zie noot 1 Deze wordt normaal gebruikt om het modem en de verbinding tussen de computer en het modem te testen. Wij gebruiken deze mode nu om de letters die we intikken op het beeldscherm te krijgen. Alles wat we intikken na het inschakelen van de LOG-funktie zal op het scherm verschijnen, en alles wat op het scherm verschijnt wordt geLOGd. Zo kunnen we een bericht aanmaken. Nadat het bericht klaar is tikt u CTRL,M om de LOG funktie te beeindigen. Het LOGgen wordt dan gestopt, en er wordt een END of FILE-marker achter gezet. (#0D, gevolgd door #FF)
Als we hierna bijvoorbeeld ATOMTEL bellen, en op de juiste plaats ons bericht verzenden, zal de telefoonrekening de volgende keer weer iets lager uitvallen. Waarschijnlijk heeft de oplettende lezer al bemerkt dat de keus van de END of FILE-marker gelijk is aan het kenmerk dat onze ATOM achter zijn BASIC-programma's zet. Dit betekent dus dat een BASIC-programma, dat vanaf #5000 (lees het begin van het LOG geheugen) wordt geladen ook op deze manier verzonden kan worden! Een ieder zal het met mij eens zijn dat dit mogelijkheden biedt. Er is een aparte subroutine gemaakt die de data test op bytes die een waarde hebben die niet toegestaan is, bijv #02, #0C en #15, ook voor de 80-kolomskaart hebben bepaalde bytes of bytecombinaties soms de vreemdste gevolgen. Deze subroutine is naar behoefte uit te breiden met alle karakters die niet door de VDU gedraaid mogen worden. Wel wil ik hierbij verklaren dat de autotransmit in de eerste plaats voor berichtenoverdracht bedoeld is, en niet om programma's over te sturen.

*noot1: Wanneer u een VIDITEL-modem gebruikt kan een Local Loopback niet zonder meer tot stand gebracht worden, omdat zend en ontvangsnelheid niet gelijk zijn. U dient in dit geval het modem te verwijderen, en bijv met een speciaal gemaakte stekker de zender en de ontvanger aan elkaar te knopen.

Leo Gijssel, Regio Zeeland



Diskcat aangepast voor gesimuleerde Atomdos-schijven (CP/M)

Omdat ik na aanschaffing van de Z80-kaart al mijn programma's op CP/M geformatteeerde schijven heb overgebracht kwam ik tot de ontdekking dat ik DISKCAT niet meer kon gebruiken. Daarom heb ik DISKCAT aangepast. Voor de gebruiker is het van belang dat in het programma de printercode's en vdu-statements worden aangepast aan het eigen systeem.De gebruikte printercodes zijn voor een Microline 80 printer. De VDU-statements zijn VDU80 en VDU40. Zonodig link adressen gebruiken.

VDU-statements aanpassen op regels 100,335,350,420 en 830

Printer codes aanpassen op regels 570,630,650,835,1050,1085 en 1140

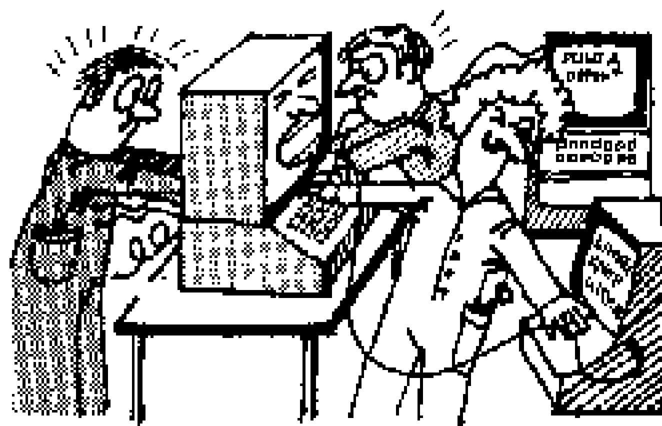
Het programma was al verder aangepast door Berry Lam, deze heeft de losse file's al achter elkaar gezet zodat niet telkens verschillende file's geladen hoeven te worden. Ook heeft hij de sorteerroutine aangepast en ingekort. Hartelijk dank hiervoor.

Er zijn nu nog 3 files nodig :Diskcat;DCMCass;FCAT . De vierde file:startDCMcass wordt alleen gebruikt bij eerste opstart.
het hoofd programma moet geladen en gerunt worden op #8200. Het programma DCMCass wordt geladen op #2700, hier dient men dus wel geheugen te hebben ,anders moet men het relocaten naar een andere plaats. Ook is de 80-kolomskaart nodig

Voor informatie over het programma zelf verwijs ik naar vorige beschrijvingen.

Veel genoegen bij het gebruik.

B.Meintema
Oud Ambacht 131
9201 VP DRACHTEN



A T O M M A R K T

Te KOOP aangeboden :

computer : ACORN ATOM in originele behuizing met:
ingebouwde Combikaart + 80-koloms-clubkaart
RAM van #2000 - #9FFF en van #8400 - #8800
4 EPROM-blokken + 2 RAM-blokken op #A000
2 RAM-blokken van #1000-#2000
ingebouwde accu voor Write Protect
voeding : Professionele geschakelde voeding
Fabrikaat Klaasing 5V/5A ; 12V/3A
Monitor : AVT, groen, bijzonder fraai beeld
Drives : 2 stuks TEAC drives 40-track SS
Originele ATOM-controller
Alles geplaatst in originele ATOMDOS-behuizing
Floppy's : ± 70 stuks + floppy-box
Diversen : Reserve onderdelen zoals nieuw toetsenbord, nieuwe kast , IC's, EPROMS , handleidingen, documentatie-materiaal etc.

De computer is zeer zorgvuldig gebouwd met professionele IC-voeten, werkt dus perfect en is uiterst betrouwbaar. De 80-kolomskaart en de AVT-monitor zorgen voor een perfect beeld.

Prijs fl. 750,= , verkoop alleen compleet.

Jan Bronzwaer , Plautusstraat 18 6417 VJ Heerlen, tel. 045-719647

Te KOOP aangeboden :

Originele ATOM-DRIVE, zonder controller,
met schema (ook van de controller).

Prijs fl. 50,=

Jelle Aarnoudse , tel. 045-710139

Te koop gevraagd:

Standaard Atom (dus zonder randapparatuur en uitbreidingen)

R. Leurs

Nattenhoverkoestraat 5

6129 LE Nattenhoven

Tel. 04450-36454

Te koop gevraagd:

Disc-drive 40 tracks (Shugart-aansluiting)

GEEN orginele Atomdrive

Rick Crutzen

Tuinstraat 42 CJ Echt

Tel. 04752-2350

Wilt u lid worden van de ATOM COMPUTER CLUB?
Neem dan contact op met de penningmeester van de regio waar u bij ingedeeld wilt worden. Deze kan u inlichten omtrent het lidmaatschap.

Regio OVERIJSSEL/GELDERLAND;
Zie regio Arnhem

Regio TWENTE;
G.J.Noorland Prinses Ireneweg 4 7433DE Schalkhaar
05700-25294

Regio NOORD-HOLLAND;
P. van Kuik Zuideinde 54-a 1843 JP Groot Schermer
02997-1902

Regio DEN HAAG;
Th.Waayer L.Couperusstraat 6 2274XP Voorburg
070-862504

Regio DELFT;
Zie regio Den Haag

Regio ROTTERDAM;
Zie regio Den Haag

Regio CENTRUM;
P.van Mourik Ruiterstede 60 3431 XN Nieuwegein
03402-48781

Regio ARNHEM;
J.Hartog Keyenbergseweg 60 6871 WK Renkum
08373-13757

Regio ZEELAND;
E.Gijssel Ruysdaelstraat 6 4462 AD Goes
01100-32557

Regio BRABANT-OOST;
F.Ehrlig Roostenlaan 266 5644 BS Endhoven
040-114183

Regio LIMBURG;
J.Colen Provincialeweg Z-27 6438 BA Oirsbeek
04492-1957

Regio BELGIE;
Zie regio Limburg

Bij het aangaan van het lidmaatschap kunt u de contributie overmaken op de rekening van de federatie.Vermeld hierbij uw volledige naam, adres en de regio waar u bij ingedeeld wilt worden.